JPRS L/9478

12 January 1981

USSR Report

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 1/81)

FBIS FOREIGN BROADCAST INFORMATION SERVICE

NOTE

JPRS publications contain information primarily from foreign newspapers, periodicals and books, but also from news agency transmissions and broadcasts. Materials from foreign-language sources are translated; those from English-language sources are transcribed or reprinted, with the original phrasing and other characteristics retained.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by JPRS. Processing indicators such as [Text] or [Excerpt] in the first line of each item, or following the last line of a brief, indicate how the original information was processed. Where no processing indicator is given, the information was summarized or extracted.

Unfamiliar names rendered phonetically or transliterated are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear in the original but have been supplied as appropriate in context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by source.

The contents of this publication in no way represent the policies, views or attitudes of the U.S. Government.

JPRS L/9478

12 January 1981

USSR REPORT

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 1/81)

CONTENTS

HARDWARE

UDC 681.31.324

SPECIALIZED PROCESSORS

Kiev SPETSIALIZIROVANNYYE PROTSESSORY. INSTITUT ELEKTRODINAMIKI. PREPRINT 224 in Russian signed to press 11 Mar 80 pp 1-2, 61-64

[Annotation and Table of Contents from Preprint 224 of the Institute of Electrodynamics of the Ukrainian Academy of Sciences, printed in accordance with the decree of the Scientific Council of the Electronics and Modeling Sector of the Institute of Electrodynamics of the Ukrainian Academy of Sciences, edited by V.N. Polozhintsev and N.N. Zinchenko, 200 copies, 64 pages]

[Text] The fundamental principles of the structural organization and software for specialized parallel processors, intended for the solution of a broad class of applied problems, are briefly set forth in the materials presented here.

Table of Contents

[31-8225]

8225
CSO: 1863

UDC 681.31.324

A SPECIALIZED CONVEYOR PROCESSOR FOR A FAST FOURIER TRANSFORM

Kiev SPETSIALIZIROVANNYYE PROTSESSORY. INSTITUT ELECTRODINAMIKI.
PREPRINT 224 in Russian signed to press 11 Mar 80 p 4

[Paper by G.M. Lutskiy and V.V Kovalenko, Kiev]

[Text] Questions of the realization of a specialized conveyor processor for a fast Fourier transform are treated in the paper, where the processor operates in real time in a frequency range of 0 to 250 KHz, and which computes 1,024 Fourier coefficients within 4 milliseconds. The calculation of ten iterations is accomplished in one conveyor arithmetic unit (AU). A 565RU2 integrated circuit semiconductor memory with an access cycle of 400 nanoseconds, which is four times longer than the conveyor waiting time, was selected as the memory (ZU). For this reason, to achieve 100 percent utilization of the arithmetic unit, i.e., matching of the operational speed of the conveyor to the memory, the latter is organizaed on the basis of matrix and conveyor principles and incorporates six memory modules with a volume of 18 x 2^{10} bits each. To store complex numbers, the memory modules are combined in blocks, with two modulses in each one. The parallel operation of the memory is realized at the module level, while the operation with the combined memory is realized at the level of the memory blocks.

The central control circuitry contains a central control signal transducer, conveyor switchers and access drivers.

[31-8225]

8225
CSO:1863

UDC 681.31.324

AN INDIRECT METHOD OF COMPUTING A FOURIER SPECTRUM USING A SPECIALIZED CONVEYOR PROCESSOR

Kiev SPETSIALIZIROVANNYYE PROTSESSORY. INSTITUT ELEKTRODINAMIKI. PREPRINT 224 in Russian signed to press 11 Mar 80 pp 6-8

[Paper by A.A. Pokuchayev, V.A. Zentsov, S.F. Svin'in and V.B. Smolov, Leningrad]

[Text] An indirect method, which is an alternative to the direct method of computing the Fourier spectrum of an analog signal, as well as the device for its realization, are analyzed.

It is well known that direct computation of a Fourier spectrum using the fastest of the existing methods, a fast Fourier transform algorithm (BPF), entails the necessity of processing complex numbers, and in particular, the execution of the multiplication of two complex numbers. For this reason, despite the possibility of conveyor processing of analog signal readouts, the operational speed of such a processor is limited for the execution of a BPF.

The proposed method consists in the rapid calculation of the output signal spectrum in some intermediate base of functions and in the subsequent calculation of the calculated spectrum to thr Fourier spectrum by means of multiplication of the column vector of the intermediate spectral coefficients by a special matrix for the transformation of the spectra.

A base of piecewise linear functions, which are the primitives of Walsh functions, was selected as the intermediate base of functions. It can be demonstrated that the rate of calculation of an analog signal spectrum in a base of these functions if practically equal to the rate of spectral calculations in a base of Walsh functions when using a fast Walsh-Fourier transform algorithm, which, as is well known, requires significantly less time for the processing of a fixed number of readouts, as compared to a fast Fourier transform algorithm.

The computation of the signal spectrum in a piecewise linear function base consists in applying the fast Walsh-Fourier transform algorithm to the first right differences of the input signal readouts. In this case, the precision in the calculation of the desired analog signal Fourier spectrum is successfully increased as compared to the fast Fourier transform method through the piecewise linear interpolation of the input signal during the calculation of the intermediate spectrum.

The specialized processor which realizes this method consists of an analog-digital converter, a block for calculating the first right differences, a fast Walsh transform processor and an arithmetic unit with a permanent memory (PZU) and an optical memory (OZU) [sic] included in its complement.

The method treated here makes it possible to parallel the computational process in each stage of the calculations. Following the conversion of the analog input signal values to a digital code, the block for calculating the differences determines the values of the first right differences. The fast Walsh transform processor, based on the readouts obtained, simultaneously calculated all 2^n Walsh coefficients. It is significant that during the sequential arrival of the readouts of the differences at the inputs to the BPU [fast Walsh transform] processor, conveyor processing of the original values is accomplished. If the readouts of the differences are fed to the inputs of the BPU processor simultaneously, then they are processed using an iteration algorithm over n iterations. The arithmetic units performs the parallel multiplication of the column vector of the spectral coefficients by the rows of the conversion matrix for the spectra stored in the permanent memory. The resulting spectral Fourier coefficients are simultaneously written into the immadiate access memory for further processing.

The conveyor structure of the processor (K-structure) makes it possible to significantly increase the performance with respect to a structure having one flow of instructions by virtue of a slight complication of the readout flow control algorithm. A further performance increase takes the approach of changing over to an iterative processor structure.

[31-8225]

8225
CSO: 1863

UDC 681.31.324

A STRUCTURAL APPROACH TO THE DESIGN OF HARDWARE AND SOFTWARE FOR SPECIALIZED MICROPROCESSOR SYSTEMS

Kiev SPETSIALIZIROVANNYYE PROTSESSORY. INSTITUT ELEKTRODINAMIKI. PREPRINT 224 in Russian signed to press 11 Mar 80 pp 16-17

[Paper by V.Yu. Grigor'yev, Moscow]

[Text] The utilization of microprocessor (MP) BIS [large scale integrated circuit-LSI] for the realization of specialized processors (SP), computer devices which are oriented in terms of the programming and hardware towards the most effective solution of a rather group of problems, is placing new requirements on the design of specialized microporcessor systems (SMPS).

A considerable expansion of the area of application of specialized processors, first of all because of the reduction in hardware expenditures, insistently requires a reduction in the timeframes and cost for the development of both the architecture and the software (MO) of SMPS's, including its set-up for a specified application.

The key to the resolution of this problem is top-down planning - a methodology which makes it possible to use the successive continuity of both the hardware and the software in the realization of a large assortment of specialized microprocessors for the solution of a broad class of problems based on program and hardware compatible sets of SMPS structures and modifiable software modules.

The utilization of microprocessor LSI's, the component base of fourth generation computers, governs the application of the "ZM" structural principle:

1) Modularity at all hierarchical levels;
2) Trunk data exchange in the system;
3) Microprogrammability of the system as a whole.

An analysis of the broad assortment of microprocessor LSI's, being series produced both by domestic and foreign electronics industry, shows that the most suitable for applications in SMPS's are micro-programmable bit-modular microprocessors, which provide for flexibility in the realization of both the hardware structure and the software of the specialized processors.

Structural microprogramming is proposed as the basic principle for the development of the software for specialized microprocessor systems: a method of direct control of the functioning of the SMPS components by means of structural sequences of micro-instruction, which take the form of an expansion of the code of the micro-operations of the central processor sections. The indicated method takes into account the major trends in the organization of contemporary control systems: structural programming as a means of increasing the productivity of the design of software for computer systems; increasing the computational capacity of the micro-operations of bit-modular microprocessors, utilizing internal microprocessing based on programmable logic matrices, and as a consequence of this, a reduction in the overall number of micro-instructions in the microprogram, which realizes the algorithm for SMPS functioning.

An example of the application of the methodsdescribed for the planning of specialized processors is the development of a series of SMPS's, designed around four-fit microprocessors with K584IK1 sections, which cover a broad range of specialized processor computer performance.

Included in specialized processors which have been developed are a parallel, dual processor digital filter for a formant speech synthesizer, digital and analog-digital models of proportional-integral-differential controllers for the control of actual physical objects and a multi-processor microprocessor model system for the solution of systems of differential equations, which can be controlled from the "Elektronika-60" microcomputer.

Developmental experience with the SMPS's enumerated above has shown a significant advantage of the structural approach over the existing practice of programming specialized processors at the level of the specialized set of instructions, oriented towards a definite class of problems.

[31-8225]

8225
CSO: 1863

UDC 681.31.324

THE STRUCTURE AND HARDWARE REALIZATION OF A HYBRID COMPLEX FOR THE MODELING OF THERMAL FIELDS

Kiev SPETSIALIZIROVANNYYE PROTSESSORY. INSTITUT ELEKTRODINAMIKI. PREPRINT 224 in Russian signed to press 11 Mar 80 pp 17-18

[Paper by Yu.P. Kamayev, Yu.N. Kolomiytsev, Yu.M. San'ko and I.A. Frenkel', Kuybyshev]

[Text] The most effective tool for the study of processes in objects with distributed parameters, in particular, processes of heat exchange, diffusion and filtration, is mathematical modeling. The modeling of nonlinear nonsteady-state thermal processes makes it necessary to increase the operatinal speed of computers.

The architecture and hardware for an analog-digital modeling system based on a parallel specialized hybrid processor is treated in this paper. The control of the processor is accomplished by a small digital computer, coupled to it through an interface.

The analog-digital processor, which is designed to model thermal problems, is based on the network principle for the representation of the initial equations. The coefficients of the network operator are specified in digital form, while the solution is obtained in the form of voltages at the nodes of the processor when completing the transient process in the analog section. In the given variant of the hybrid complex, the nonlinearity of the coefficients is acounted for by a digital computer. The possibility of designing program variable nonlinear blocks to increase the operational speed of the modeling of nonlinear problems is analyzed.

[31-8225]

8225
CSO: 1863

UDC 681.31.324

A MULTIPLE MICROPROCESSOR SYSTEM FOR DECODING A CONVOLUTION CODE

Kiev SPETSIALIZIROVANNYYE PROTSESSORY. INSTITUT ELEKTRODINAMIKI. PREPRINT 224 in Russian 1980 signed to press 11 Mar 80 pp 18-19

[Paper by R.A. Biktashev, N.N. Barlinskiy and V.B. Smolov, Leningrad]

[Text] Convolution codes, which are decodable using a Viterbi algorithm, find widespread application in providing for the noise immunity of data transmission. A single processor design of a Viterbi decoder based on microprocessor sets of the 580 and 589 series makes it possible to obtain a transmit rate in a communications channel of no more than 5 to 10 Kbit/sec, something which is not satisfactory to numerous users of systems with noise immune coding.

A Viterbi decoding algorithm can easily be represented in parallel form, something which makes it possible to use several microprocessors to increase systems performance with optimal organization of the breakdown of the algorithm into subtasks and the division of system resources. Microprocessor architecture also has an impact on the efficiency of a multiprocessor system.

The decoding of convolution codes using a Viterbi algorithm consists in comparing the received code sequence with referenced standard code sequences and in selecting from the reference sequences the one which is most correlated with the received sequence. In the case where the algorithm is represented in the form of a grid diagram for the design of a multiprocessor system, one can use as many microprocessors as there are nodes in the grid diagrams having a decodable convolution code.

Two variants of the architecture are treated having an common bus for the decoding of a convolution code with a code spacing of K = 9 and a transmission speed of R = 2/3, one of which has distributed control while the other has a controlling processor. Both variants have a modular structure and contain four and five computer modules respectively for processing the 16 nodes. If the system efficiency is defined as the ratio of its actual productivity to the overall productivity of all processors, then the efficiency of a structure with a control processor is three to seven percent higher than the efficiency of a structure with distributed control. This is explained by the special features of the deparalleling of a Viterbi algorithm, but has double the delay in feeding out the decoded sequence, which is equivalent to the arrival time from the channel of a number of symbols equal to twice the number of symbols, with respect to the length of the code grid.

To methods of transmitting intermodular messages are studied, in one of which, channel switching is used for communications between the microprocessors and the memory, while in the other, batch switching is employed. The interface between the microprocessors and the common bus is simplified in the case of channel switching, but the exchange increases. In the case of batch switching, the interface is complicated, but the exchange time is reduced. The efficiency of a system with distributed control and batch switching is 0.89, and that with channel switching is 0.73, which is equivalent to an increase in the productivity as compared to a single processor design by factors of 3.56 and 2.92 respectively.

[31-8225]

8225
CSO: 1863

UDC 681.31.324

THE PARALLEL REALIZATION OF INTEGRAL CONTOUR FUNCTIONS OF AN IMAGE IN A SPECIALIZED OPTICAL PROCESSOR

Kiev SPETSIALIZIROVANNYYE PROTSESSORY. INSTITUT ELEKTRODINAMIKA. PREPRINT 224 in Russian signed to press 11 Mar 80 pp 23-24

[Paper by L.B. Ochina, Leningrad]

[Text] The primary operation performed by an analog coherent optical processor is a Fourier transformation. It is well known that the square of the amplitude of a Fourier transform is invariant to translation of the input image, while image rotation leads to rotation of the Fourier transform.

To obtain a transform which is invariant to image rotation, the square of the amplitude of its Fourier transform must be represented in the form of integral contour functions (IKF) [1]. With this approach, image rotation leads to the effect of cyclical displacement of the IKF. The results of such a transformation is subjected to a new Fourier transformation, which is invariant to translation of the IKF, and consequently, image rotation.

It has been demonstrated in the literature [2, 3], that using optical methods, including methods of physical and digital holography, it is impossible to transform an IKF image from a Cartesian system of coordinates to a polar system. A specialized analog electronic computer [3] is used to solve this problem, the basis for which is a closed television image transmission channel. The transformation is accomplished in the image transmission channel.

In keeping with papers [4, 5], we represent the input image in the form of a matrix T_{ij}, where $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots J$. We represent the result of the transformation of this image to a polar system of coordinates in the form of a matrix W_{kl}, where $k = 1, 2, \ldots K$, $l = 1, 2, \ldots L$. Then by direct tabulation on a computer, we specify the law governing the transformation of T_{ij} to W_{kl} in a tabular form as a correspondence matrix H, having I rows and J columns. Each element of this matrix contains the matrix for the transformation of an element of the original matrix T_{ij} to the corresponding subset of elements of the output matrix W_{kl}.

The technical realization of this transformation can be based on a page oriented holographic permanent memory (GPZU). The Fourier hologram matrix with dimensions of I x J corresponds to the transformation matrix H. The Fourier transform of the page of information corresponding to the matrix H_{ij}, is stored in each hologram. A photo transparency of the image, which is the square of the amplitude of the Fourier transform of the input image, is placed in front of the hologram matrix. This photo transparency is illuminated by a plane monochromatic light beam. Two images, written in the Fourier hologram matrix, are simultaneously restored in the output plane of the permanent holographic memory. The total image, which is the result of the transformation to a polar system of coordinates, is subjected to a new Fourier transformation.

BIBLIOGRAPHY

1. Valil'yev V.I., Reutskiy V.Ye., "Opisaniye izobrazheniy integral'nymi konturnymi funktsiyami" ["The Description of Images by Integral Contour Functions"], AVTOMATIKA, 1967, No 2.

2. Kesesent D., Psaltis D., "Novyye metody opticheskikh preobrazovaniy dlya raspoznavaniya obrazov" ["New Optical Transformation Methods for Pattern Recognition"], TIIER, 65, 1977, No 1.

3. Casasent D., Psaltis D., "Position, Rotation and Scale Invariant Optical Correlation", APPLIED OPTICS, 1976, 15, No 7.

4. Smolov V.B., Popov, S.A., Li S.K., et al., "Parallel'nyye geometricheskiye preobrazovaniya izobrazheniy v opticheskikh protsessorakh" ["Parallel Geometrical Transformations of Images in Optical Processors"], Kiev, Znaniye Obshchestvo Publishers, 1977.

5. Smolov V.B., Li S.K., Ochina L.B., et al., "Model' parallel'nogo matrichnogo protsessora" ["A Model of a Parallel Matrix Processor"], Kiev, Znaniye Obshchestvo Publishers, 1977.

[31-8225]

8225
CSO: 1863

UDC 681.31.324

METHODS OF CONSTRUCTING DIGITAL NETWORKS ORIENTED TOWARDS THE SOLUTION OF DIFFERENCE BOUNDARY PROBLEMS OF LARGE DIMENSIONS

Kiev SPETSIALIZIROVANNYYE PROTSESSORY. INSTITUT ELEKTRODINAMIKI. PREPRINT 224 in Russian signed to press 11 Mar 80 pp 53-55

[Paper by Ye.A. Bashkov, V.P. Boyun, S.M. Voronoy, L.G. Kozlov, Yu. Yu.V. Ladyzhenskiy and L.P. Fel'dman, Donetsk]

[Text] When performing numerical experiments on mathematical models of processes described by differential equations in partial derivatives, it is necessary to repeatdely solve systems of finite difference equations of large dimensions. A new trend in the design of computer tools for the solution of problems of this class is related to the development of problem oriented parallel digital networks.

The success of the application of an analog processor in a hybrid computer system is governed by the flexibility of the structure and the completeness of the software complement, which should provide for the execution of the major steps of the process with minimum expenditures of manual labor and efficient utilization of the advantages of the analog-digital structure while simplifying the requirements placed on the knowledge of the user of the analog-digital computer complex of the special features of programming problems in it. Working from this, the software for the "Saturn-I" analog-digital computer complex (MOS) has a hierarchical structure with a distribution of the functions with respect to the level which assures a high degree of automation of the programming.

The primary function of the zero level programs is:

--Provide for data transmission in accordance with the requirements of the given communications channel;
--Transform the formats for the representation of data, monitor the status of the device, etc.

The first level programs are intended to assure the following:

--Make the transition from the mathematical model to the physical model and back with automated scale selection; make the transition from the mathematical function to a graphical image on paper or the screen of a CRT; monitor the correctness of the timewise processing of the physical model parameters, etc.

The calculation of the majority of the mathematical models which must be dealt with in the theory and practice of developing petroleum deposits is realized by numerical methods using powerful computers. The diversity of these models is obvious, and the design of some universal mathematical model which describes all of the oil and gas filtration processes in a porous medium which are of interest to us is extremely difficult. On the other hand, the compilation of procedures and programs for each of the given mathematical models entails considerable production expenditures for programming. For this reason, contemporary software for analog-digital computer users is built on a modular principle and belongs to the second level.

The set of software programs for the analog-digital computer system forms a unique modeling language for the solution of complex problems in mathematical physics without placing requirements of knowledge of the analog processor on the user. Data are given in the paper on the programs as well as examples of their utilization.

A survey is made in the paper of the major approaches to the structural realization of iteration methods of solving systems of different equations in digital networks (TsS). The structures of the nodal processors of simple iteration digital grids are analyzed, where the desired solution is computed in one nodal processor at one point in the spatial network region. Comparative estimates are given for the equipment expenditures and the operational speed of two dimensional digital networks for the solution of nonlinear elliptical differential equations with nodal processors, oriented towards methods of simple iteration and upper relaxation. Methods of constructing nodal processors which compute the value of unknowns in several adjacent points of a region are studied, which are based on the method even-odd iteration and a multiple network relaxation algorithm. A number of algorithms for bypassing the nodes of a network region are analyzed for such networks, which make it possible to simplify the organization of links between the processors and curtail the problem solving time. The modular-parallel solution of large systems of difference equations, organized in the proposed networks, also makes it possible to reduce the equipment expenditures through curtailing the number of nodal processors, as compared to the case of simple networks.

The possibilities of reducing equipment expenditures by means of organizing the computational process in a digital computer-digital network complex using the method closing errors are analyzed.

The results of experimental studies of the proposed algorithms and structures using the method of modeling on universal digital computers are given.

[31-8225]

8225
CSO: 1863

UDC 681.3:62-184.4:62-52

'VIDEOTON' REMOTE PROCESS TERMINAL MICROPROCESSOR FOR ASUTP'S

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 8, 1980 pp 3-7

[Article by E. Sarkozi and I. Grosz, Hungarian People's Republic]

[Excerpt] General structure of an RPT-based ASUTP

The "Videotor" computer plant started the development and production of computer-based ASUTP's [avtomatizirovannyye sistemy upravleniya tekhnologicheskim protsessom--automated systems for the control of technological processes] in 1971. Initially made with integrated circuits of the DTL (diode-transistor logic) series, starting in 1972 those systems were made with integrated circuits of the TTL (transistor-transistor logic) series. On the basis of experience accumulated in the process of redesigning ASUTP's, the further development and expansion of the family of devices for communications with the object continued. Characteristic of those systems was a centralized structure. Devices for communication with an object [ustroystvo svyazi s ob"yektom--USO] are placed a small distance from the computer (a maximum of about 120 m). All signals from the controlled technological process must be transmitted to the central computer by means of cables of different length and remote pickups. For that purpose the USO's were connected to a programmed parallel data channel (the so-called real-time line); the central computer program controlled their work.

Shortcomings of centralized technological process control systems constructed on the basis of passive USO's are complexity of the central computer software, which includes both USO control subroutines and a complex of consumer programs which satisfy various requirements of users, the overloading of the central processor with USO control programs and, consequently, the inadequacy of time to solve problems of a higher level, the great probability of complete halting of the system due to failure of some single unit, which greatly reduces the reliability of the system, and fairly high expenditures connected with the great length of cable needed for USO connection.

The appearance of microprocessors in 1974 brought about revolutionary changes in the development of technological process control systems. Thanks to the fact that the possibility arose of creating decentralized systems, free of shortcomings intrinsic to centralized systems and, in addition, having the following merits: a greater velocity of the system as a whole, explained by distribution of tasks between the central computer and terminals on the basis of microprocessors; the economic character of the use of microprocessors to solve simple problems where it is possible to dispense with costly mini-computers, the possibility of automating territorially separate technological processes, which earlier was a task difficult to perform, and increased flexibility of the system since the programs of the terminals can readily be changed.

In recent years a microprocessor decentralized technological process control system has been worked out at the "Videoton" plant. Its basis is the Remote Process Terminal (RPT), which has a number of distinctive characteristics.

An RPT microprocessor represents a terminal since, in spite of the fact that there is a possibility of its autonomous use, all the merits of the RPT are displayed only when it is used in a system containing a high-level computer which accomplishes connection of the RPT as a terminal. The main difference of the RPT from ordinary terminals consists in the fact that that device accomplishes primarily connection with all sensitive elements, sensors and actuating mechanisms intended for monitoring and control of a technological process, but communication with the operator is secondary or completely lacking. The application of the RPT is most effective when the distance between it and the central computer amounts to several hundred meters or kilometers, although the installation of the equipment under consideration is possible also near the computer. The exchange of information between the RPT and the computer occurs through the consecutive data transmission channel.

The three indicated circumstances also determined the name given the device: the Remote Process Terminal.

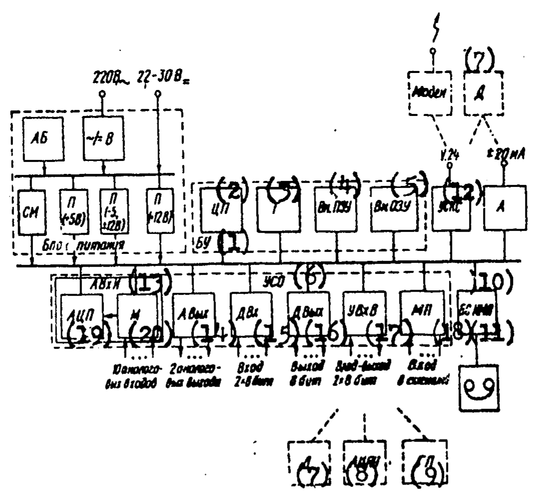


Figure 1

The structure and set of RPT modules are presented on Figure 1, where the main RPT set is distinguished. The most important element of the latter is the programmable microprocessor control unit BU (1), connected to the parallel input-output line, called MICROBUS. The control unit includes a central processor TsP (2), a programmed timer T (3) and main OZU (4) and permanent PZO (5) storage modules.

Attached to the same bus are devices for communication with an object USO (6) and devices for the coupling of peripherals and data transmission channels. The set of USO modules contains the modules described below for the input-output of analog and discrete signals, the number of which can vary in accordance with the requirements of a specific object.

To provide communication with the operator it is possible to use a display D (7) of the "Videoton" type, working in an asynchronous regime and connected to the RPT by means of a synchronous-asynchronous communications channel coupler. In that case in the reprogrammed storage of the coupler there is a program necessary for servicing the display.

A mozaic type DZM-180 alphabetic-digital printer ATsPU (8) is used to print documents, connected in parallel with the RPT by means of a universal USO input-output.

If it is necessary to obtain a graphic depiction of the working curves, characteristics, etc, an NE-2000 two-coordinate X-Y graph plotter (9) is used, connected to the RPT similarly to the alphabetic-digital printer ATsPU.

The PRT set can include a DCD-1 cassette magnetic tape store NML (10) which, when failure occurs in the data transmission channel, serves as a transitional memory for storage of very important data and their subsequent transmission. That storage is convenient to use for the loading and storage of programs in the process of adjustment, and also the loading of test programs during RPT monitoring and maintenance. Provided for connection of the magnetic tape sotre is a special device (the magnetic tape store coupler BS NML--11), which has its own microprocessor of the "Intel 8080" type, a PZU (2 Kbytes) and an OZU (1 Kbyte). In the work of the NML a channel of direct access to the RPT memory is used, and the central processor only initiates the exchange of data units. One coupler serves for the coupling of four store cassettes.

It was pointed out above that the RPT is primarily intended for use in a computer-based ASUTP as a terminal, but its technical parameters provide the possibility of autonomous application. It is used as a terminal in mining affairs, the oil and gas extraction industry, gas and water pipelines, electrical transmission lines, at industrial enterprises distributed over a large territory, in the water economy and in metrology.

The autonomous application of RPT is most advisable for laboratory measurements, the control of machine-tools, final inspection of articles and the regulation of technological processes.

The microprocessor control unit

Figure 2 presents a structural diagram of the module of a central processor TsP (1). Used as a central processor is the "Intel 8080" mocroprocessor. In addition, on its plate there is a 3-Kbyte PZU module of the REPROM type (2), the content of which is erased by means of ultraviolet radiation (the PZU is made of three 1-Kbyte REPROM integrated circuits), a 1-Kbyte OZU module of the RAM type (3) on KMOP RAM integrated circuits, 4 x 56 bits each (in case of use of a battery

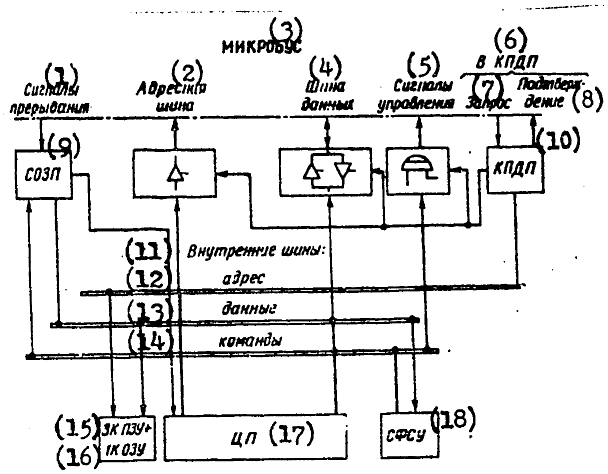


Figure 2.

Key: 1 -- Interruption signal
2 -- Address bus
3 -- MICROBUS
4 -- data bus
5 -- control signals
6 -- in KPDP (circuit for requesting channel for direct access to memory)
7 -- request
8 -- confirmation
9 -- SOZP (circuit for processing interruption requests)
10 -- KPDP
11 -- Internal buses:
12 -- address
13 -- data
14 -- instructions
15 -- 3K PZU (reprogrammable memory)
16 -- 1K OZU (main memory)
17 -- Central processor (TsP)
18 -- SFSU (circuit for formation of control signals)

power unit the content of the memory is not lost when the network voltage disappears) the circuit for processing interruption requests SOZP, a circuit for requesting a channel for direct access to the memory KPDP and a circuit for the formation of control signals SFSU. Besides main-level signals the system can process eight priority interruption requests. This is achieved because the zero level of interruption (the highest) and the initial reset can differ in the read-out of the word of state (in both cases the microprocessor starts work on the program with the initial memory address 0000). Thanks to the corresponding programming of the special register, in which the level of the program being performed is recorded, such a handling of the interruption request of the lowest level is assured.

Although the "intel 8080" microporocesor has in principle a logic serving for the realization of direct access to memory, in the RPT on the basis of ordinary integrated circuits (logical circuits, inverters and triggers) a special circuit was planned which provides improvement of the characteristics of the KPDP channel (for requesting direct access to the memory) as a result of reduction of the waiting time for direct access; increase of the system velocity thanks to reduction of the influence of operations of the KPDP channel which caused slowing down of the work of the processor.

When a request is received, only then is the work of the microprocessor established, when operations follow data exchange through the MICROBUS.

If the microprocessor works within the plante of the central processor, the received request for the KPDP channel does not halt the work of the processor but only disconnects it from the MICROBUS line. Such a situation arises rather frequently, as the very important and frequently working programs (the monitor modules, timer, recurrent subroutines, etc), and also the stack is arranged in the memory available on the central processor plate. It is natural that in such a decision for that region of the memory there is no possibility of access by means of a request for the KPDP channel through the MICROBUS line. However, that is a merit rather than a shortcoming, as it reduces the probability of random damage of the content of the designated memory. Therefore besides the stack (which in any case requires protection) it is advisable to arrange the most important data in that region of the memory.

The logical circuit for formation of a service signal order issues a signal by means of which the work of the microprocessor can be halted by feeding an appropriate signal on the part of the external connector. When the service unit is used there is a possibility of step-by-step halting of the work of the microprocessor with respect to address.

The control unit also includes (see Figure 1) a programmed timer T, which has a counter, the content of which and the reset periodicity are established according to the program. The cadence signal of the counter arrives from a quartz generator (with a relative error of less than $5 \cdot 10^{-5}$); the interruption request is formed when the counter is reset.

The established periodicity of formation of the request signal is from 1 millisecond to 15 seconds. In addition, on the timer T plate provision is made for circuits serving to provide the possibility of connecting the operator's panel, and also circuits for the formation of signals for request for highest-level interruption. Reasons for disappearance of the interruption request, the non-sending of a response signal, switching off the power of the KMOP RAM type main memory (for example, battery discharge) and restoration of the network voltage.

The RPT memory can be expanded to 65 Kbytes by connection to the MICROBUS line of external memory units (in addition to the memory units on the central processor plate). In that case the reprogrammable and main storage volumes can be in any ratio. The maximum volume of a PZU module of the REPROM type is 16 Kbytes; expansion is made in 1 Kbyte intervals. The volume of a single OZU module of the $\text{REPROM}_{\text{ext}}$ type is 4 Kbytes. There are two types of integrated-circuit main memories: the MOP RAM and the KMOP RAM. When a KMOP RAM is used, long storage of information is assured by means of a nickel-cadmium battery even when the network voltage disappears.

In the RPT system there are three methods of systematic data transmission.

1. Data transmission at a relatively high rate (50-600 buads) on physical lines. For that purpose an adapter A of telegraph communication channel ($\pm$ 20 mA) with galvanic uncoupling has been developed. One adapter serves for the connection of one duplex communications channel.

2. Data transmission at a medium rate (600-9600 bauds) over telephone or other equivalent communication channels. The interface corresponds to assumption V. 24 according to the MKKTT [not further identified]. Synchronous and asynchronous data transmission regimes are possible. The synchronous-asynchronous communications channel coupler USKS (12) contains its own "Intel 8080" microprocessor and also a 2-Kbyte PZU module of the REPROM type and a 256-byte OZU of the RAM type. Thanks to this any data transmission algorithm can be used, having accomplished for that purpose a simple exchange of the REPROM PZU content. The device works using the KPDP channel and an independently realizing data transmission algorithm. In that case the central processor CP (2) must only know the address of the memory region where the data intended to be transmitted are stored, and also the memory in which the arriving information will be recorded. One USKS device serves one duplex channel.

3. A high-speed circuit with a coaxial cable (75 m) with a speed of about 500 bauds is in the planning stage at present. The conclusion of development is expected at the start of 1981. In that case the topology of the system will represent an open ring, the number of connected stations is about 31 + a dispatcher line, and the maximum distance is about 2000 m.

Device for communication with an object (USO)

The set of USO modules of an RPT terminal (see Figure 1) contains an analog integrating input AVkhI (13), an analog output AVykh (14), a discrete optron input DVkh (15), a discrete relay output DVykh (16), a universal input-output UVkhV (17) and an optron interruptions multiplexor MP (18).

The principal characteristics of USO modules are examined below.

The AVkhI (13) analog integrating input functionally consists of two parts arranged on two plates: a multiplexor M (20) and an analog-digital converter ATsP (19). The multiplexor is made of a relay with a programmed amplifier; voltage or a current signal can be fed to any of 10 multiplexor channels (in the latter case precise resistances are soldered to the input). After an input RC-filter and a special multiplexor relay the signal arrives at the input of the programmed amplifier, which has an amplification factor set earlier at 10, 100 or 1000. Through an analog key the amplified signal arrives at an analog line. Signals from several multiplexors can arrive over that line. The signal is fed from the analog line to the ATsP analog-digital converter.

A distinctive feature of the module is the presence of a circuit to compensate zero shift of the programmed amplifier, the working principle of which is as follows. Before integration is started the analog key short-circuits the amplifier input. In that case a voltage corresponding to the zero drift error appears on the amplifier output. That signal is remembered by the analog memory. In the process of integration, formed with a useful signal (with the opposite sign), the error approximately compensates itself.

To reduce the relative measurement error the range of measurements is automatically switched in the analog-digital converter, thanks to which the measured signal always is in the upper part of the given range of measurements. The integration time is adjusted to the frequency of the network power voltage at the given moment by means of a special circuit which establishes the cycle generator frequency in accordance with the network frequency. Galvanic uncoupling of the analog input is assured by isolated converters P of direct current into direct current, and also optrons (photodiode elements which assure galvanic uncoupling).

Principal technical characteristics of an integrating analog input

Input signal range:	
voltage in V	$\pm$ 10 and $\pm$ 100 mV, $\pm$ 1 and $\pm$ 10 v
current in mA	$\pm$ 5, $\pm$ 20
Relative error at 20°C	Less than $5 \cdot 10^{-4}$
Temperature stability, ppm/°C	More than 100
Minimum noise suppression coefficient, dB	
normal type at 50 Hz	40 (without input filter) and 60 (with filter)
general type	70 (in range of $\pm$ 10 V) and 90 (in other cases)
Maximum allowable voltage of general form in relation to computer zero, V	$\pm$ 100=
Number of measurements per second	Up to 10

The analog output AVykh has on one plate two galvanically uncoupled channels which assure voltage or a current signal on the output.

Principal analog output technical characteristics

Output signal range:	
voltage in V	0-10, $\pm$ 10
current in mA	0-5, 0-20
Resolution in digits	12 + 1 sign
Error of conversion at 20°C in low-order units in the case of:	
output in voltage	Less than $\pm$ 1
current output	Less than $\pm$ 1.5
Temperature stability in ppm/°C in the case of:	
output voltage	More than 100
current output signal	More than 150
Maximum allowable potential difference with respect to computer zero, in V	$\pm$ 100=

Each optron discrete input module DVkh receives 2 x 8 bits with the possibility of external synchronization. The input signals level is 0-4 V (logical "0"); 10-30 V, maximum 10 mA (logical "1"). The maximum allowable potential difference with respect to computer zero is $\pm$ 250 V=.

The discrete relay output DVykh has eight two-position relays and occupies two positions on the MICROBUS plate of RPT equipment. The relays permit switching the volatage to 100 V of direct current or switching current circuits to 2 A of direct current with a power of up to 100 V·A. The switching rate is about 200 bytes/second. The allowable potential difference is about ± 250 V of direct current (in relation to computer zero).

The universal input-output UVkhV serves for the connection of peripherals to the RPT terminal and assures the work of a bilateral data transmission channel (2 bytes) with programmed signal synchronization. The level of the transmitted signal matches the level of signals for TTL integrated circuits.

The MP multiplexor is intended for the transmission to the microprocessor TsP of up to eight galvanic uncoupling external interruption signals with the use of a single level of the MICROBUS line (in accordance with the established priority signals). The signal levels are the same as those of the DVkh module.

The RPT power unit has two distinctive features. Firstly, it is capable of working from an alternating current network with use of a rectifier V or from a direct-voltage source of 24 V (22-20V), thanks to which the possibility of powering the terminal from a battery is assured. A need for that arises when failurefree work is required or when the RPT is operated in a place where there is no power network (for example, on some means of transport). Used to obtain the necessary potentials are direct-current voltage converters P with galvanic uncoupling. Secondly, an adjustable buffer battery can be additionally used in the power unit; a special battery with a large load capacity (nickel-cadmium) is used for that purpose.

The latter assure maintenance of the power supply of the entire equipment for 5 minutes, and in case the network voltage disappears produces reports to the central processor about the removal of power, issues instructions in accordance with the emergency situation and maintains the work of the RPT without failure until the network voltage is restored.

Design of the RPT

The RPT microprocessor is made in a casing about 480 mm wide and 180 mm high. The printed wiring panel in which plugs a mounted for the attachment of plates forms two racks in the casing. The RPT casing can be mounted in a desk-type cabinet 650 x 450 x 240 mm in size or in a stand. Set on one rack are a power unit and a cassette type magnetic tape store, and on another--a processor, modules for the coupling of peripherals and a USO. The rear panel has a bilateral printed wiring of the main-line input-output circuit; the two sides of the panel are connected by a plate with connecting joints. There is no need to use cables, as all the signals pass through printed wiring of the rear panel, in which the RPT plates are installed.

The device was constructed in the form of modules: those arranged in 31 positions of plates are filled with various modules and in accordance with the requirements and tasks of the system.

One can connect to a single terminal analog inputs (120 channels) and outputs (24 channels), discrete inputs (320 bits) and outputs (a contact relay, 96 bits), and interruption inputs (96 channels).

The operating conditions are: a temperature of 5-40°C, a maximum relative humidity of 90 percent (at 30°C) and an air pressure of 84-107 kPa.

[23-2174]

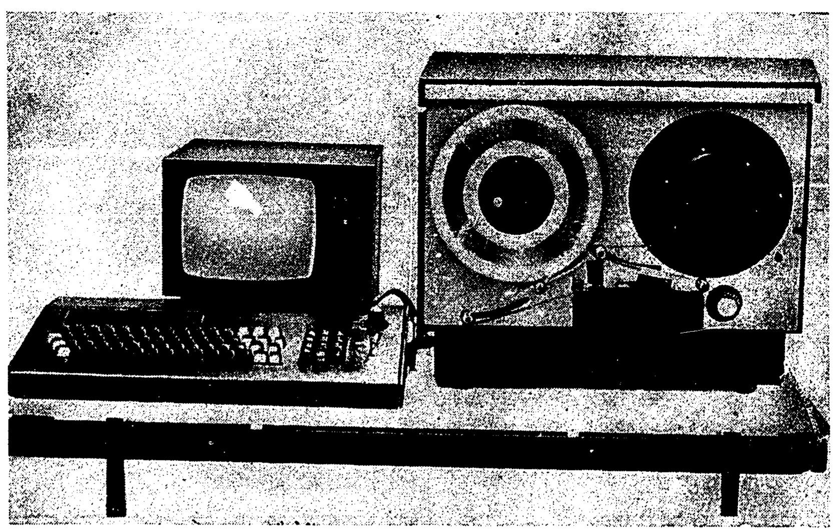
2174
CSO: 1863

YES-9004 DEVICE FOR PREPARATION OF DATA ON MAGNETIC TAPE

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 8, 1980 p 49

[Advertisement of VTO Isotimpex, Sofia, ulitsa Chapaeva, 51. Telephone: 73-61; Telex: 022731, 022732]

[Excerpt]



Technical Characteristics

Recording density, bits/mm	32
Recording method	RZ-1
Tape velocity, cm/s	39.6
Buffer storage capacity, bytes	160
Unit length, bytes	80 & 160
Recording format	ISO

Main operating modes	Data input, checking and retrieval
Other modes	Program input, program checking, printing and input from punched-card counter
Programs	2 independent
Indication	unit
Power supply, V	220 (+10%, -15%)

[23-2174]

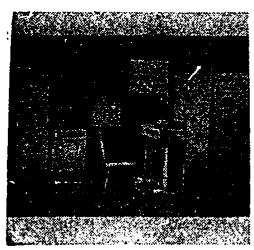
2174
CSO: 1863

NEW INSTRUMENTS AND MEANS OF AUTOMATION

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 8, 1980 pp 23-26

UDC 681.17

CENTRALIZED MONITORING MACHINE OF THE M-40-43 SERIES



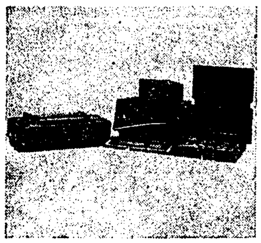
[Text] The machine is intended for the solution of problems in monitoring and controlling technological processes at enterprises with a continuous and discrete character of production and represents a further development of machines of the M-40 series, for which the principle of microprogrammed control hsa been set as a basis.

Distinctive features of machines of the M-40 series are better working reliability, simplicity of servicing, a high interrogation rate (up to 1500 signals/second) and primary data processing (500,000 operations/second), the possibility of work with objects up to 3 kilometers distant, high precision of conversion of analog signals (0.3 - 1 percent), noise immunity of the information channels and a total of up to 150 dB of the measurement track. The M-40 machine gathers information on the course of a process, its processing, the formation of starting information for carrying out a technological process, the presentation to the operator of information about the technological process at the control panel, an alphanumeric terminal and printer and the issuance of signals of two-position regulation. The M-40 machine is issued in four models. The capacities of the external main memory and the main and permanent storage units are 16K, 2K and 8K bytes respectively. The maximum number of analog input signals is about 512, and of points of discrete information input is about 1400. The maximum rate of execution of microinstructions is 500,000 operations/second.

The machine is produced by the Moscow "Energopribor" Experimental Plant.

UDC 658.012.011.56:681.321

THE "ISKRA-1269" ELECTRONIC KEYBOARD COMPUTER, PROBLEM-ORIENTED TOWARD SOLUTION OF INTERACTION PROBLEMS IN AUTOMATED SYSTEMS FOR CONTROL OF ELECTROCHEMICAL EXPERIMENT



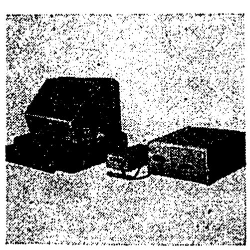
[Text] The machine was created for calculation of optimum experiment parameters, the formation of information for control of an experimental installation, the gathering and primary processing of data during experiment control according to a prescribed program, the effective depiction on a cathode-ray tube of data in a symbolic-graphic representation and the process of conducting an experiment and during subsequent processing, effective data input by means of a light pen and keyboard, documented by textual and graphic information.

The input language is BASIC, with graphic interaction operators. The machine accomplishes program input and realization, the execution of operators and instructions during manual conditions and program adjustment. It performs arithmetic operations, elementary functions and operations of the construction and transformation of graphic objects. The range of representation of numbers is from $1 \cdot 10^{99}$ to $(1-1 \cdot 10^{-11})10^{99}$. The mean time required to perform the operations of addition and subtraction is 0.6 ms, of multiplication and division is 2 ms and of elementary functions is 5 ms. The main sotre capacity is 12K bytes. The textual information format is 16 lines with 63 characters per line. The graphic information format is $2^8 \times 2^8$ addressable points. The light pen introduces and traces the marker, indicates an element of the graphic image and makes erasures and drawings. The power consumption is 600 W.

The system was developed by the State Union Design and Technological Office for Computer Planning, Leningrad.

UDC 681.327.036

THE "ISKRA-325" ELECTRONIC MONITOR AND REGISTER WITH AUTOMATIC DATA READ-OUT FROM TAGS AND DATA REGISTRATION ON MAGNETIC TAPE IN A MINI-CASSETTE



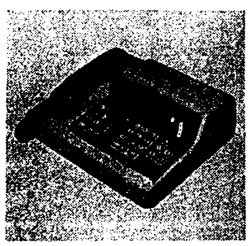
[Text] The machine is intended for the mechanization and automation of the formation, read-out and monitoring of commercial operations and accounting units of department stores with large food departments (self-service department stores). It is used as a peripheral in automated systems for management of commercial enterprises where it is used to perform tasks in the counting of cash receipts, fulfilment of the plan for commodity turnover and the sale of commodities, the preparation of statistical reports and analysis of receipts from the sale of remnants to organize retail trade, keep records of credit and study and forecast demand. The machine permits the introduction of information about a commodity from a machine-read language and a keyboard, calculates the amount of a purchase by multiplying the price by the quantity, the introduction from the keyboard of the sum of advances for credit, the sum of the percentages for credit and commissions, differentiated recordkeeping of moneys by sections, recordkeeping of moneys for cancellations or returns of goods, calculations of amounts due to purchasers and change, addition, subtraction and multiplications of whole numbers when introduced into the keyboard.

The machine has 15 counters, the printing rate is 2.5 lines/s, the power consumption is 200 W and the machine has dimensions of 485 x 405 x 400 mm.

The machine was developed by the State Union Design and Technological Office for Computer Planning, Leningrad.

UDC 681.3.01

THE "ISKRA-900" TERMINAL WITH VERY SIMPLE DATA PROCESSING



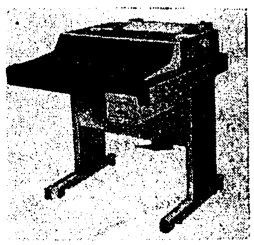
[Text] The machine is intended for the solution of problems in very simple arithmetic data processing, keeping records of time, the automation of the installation of telephone network connections, the simultaneous conducting of conversations between three telephone network subscribers, the exchange of digital information with a computer and orientation toward the use by middle and upper management of enterprises and institutions in various spheres of the national economy, preparers of data from remote ASU points, and in some applications also directly by production personnel. The terminal can work automously and jointly with computers of the M-6000 and SM series. The combination of the above functions in the terminal permits dispensing with a group of independent single-function instruments with corresponding purposes (electronic keyboard computers, calendar clocks, telephones with automatic dialing and teletypes) and in isolated cases reducing the number of operations with data during their preparation for input into a computer. Transmitted and received data are recorded on the indicator panel. The dimensions of the machine are 360 x 300 x 120 mm. It mass is 6 kg.

The machine was developed by the State Union Design and Technological Office for Computer Planning, Leningrad.

UDC 681.327.11

THE RI-7501 BIS AUTONOMOUS DATA REGISTER

[Text] The equipment is intended for the collection and preliminary processing of alphanumeric data, monitoring its reliability, registration and ecchange over a communication channel with a central processing device in an ASU. It provides programmed input of alphanumeric data from a keyboard and technical carriers (punched cards, magnetic tapes in a mini-cassette, key-badges), arithmetic processing and



and monitoring the reliability of introduced data, the formation and printing of primary documents, data registration on a technical carrier (magnetic tape in a mini-cassette or on punched tape), data exchange with the central equipment over a communication channel. Provision is made for the possibility of execution in various modifications different in the set of input-output devices.

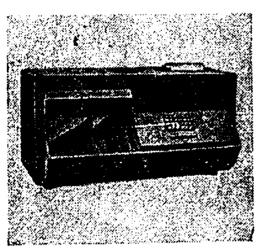
The main store volume for programs and data is 4K bytes. The number of programs is determinned by the main store volume.

The composition of the operations is: four arithmetic actions, calculation of percentages and percentage ratios, entire separation, accumulation, change of sign, comparison, conditional and unconditional transitions, input-output operations, reference to subroutines and the organization cycles. The programming language is specialized and problem-oriented, assures programming the working conditions of the register, including data input, processing, monitoring, registration and transmission. The power consumption is -280 to +340 V·A.

The equipment was developed by the State Union Design and Technological Office for Computer Planning, Leningrad.

UDC 681.327.45

THE SPK-80 MODEL Pr-22 APERTURE CARD RETRIEVAL SELECTOR

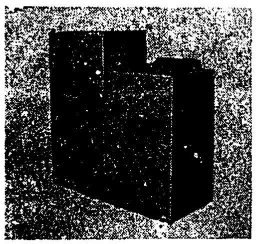


[Text] The device is designed for rapid retrieval of information placed on standard 80-column punched cards in the form of perforations. Punched cards are selected when there is complete coincidence of information on a selected punched card with that placed in the store (the regime and in the columns); coincidence of the information of a definite column of a selected punched card with respect to even one of the signs placed in the store for the same column (the regime OR in the column). It permits eliminating any number of columns from the retrieval. It is used in patent and author's certificate searches, in searches for necessary information in a technical information bureau, analogs of designs for the use of prepared solutions or study of the existing technical level of a problem to be solved, in personnel sections, libraries and other institutions. The retrieval program is entered manually from the control panel or by means of program punched cards. The punched card feeding used in the selector and the photoelectric method of information readout make it possible to reduce considerably the wear of punched cards and at the same time increase their service life. The technical rate is 400±60 cards/minute. The power consumption is 500 V·A.

The equipment is produced by the Lubny Computer Plant.

UDC 681.327.45

THE YeS-7018 PUNCHED-CARD OUTPUT DEVICE

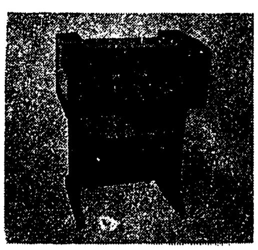


[Text] The device is intended for the receipt of information transmitted from computers in the form of electric signals, its transformation into corresponding perforations on 80-column punched cards. It is connected to the multiplex or selector channel of a YeS computer or any other computer if coupling is possible. It works in multiplex and exclusive regimes. It has an electromechanical operating principle. It provides output of information on punched cards with hardware conversion of code DKOI and punched-card code KPK-12, and also in any code without hardware conversion. It performs the following instructions: "verify input-output," "determine state more precisely," "idle running," "feed and write without code conversion," "control," "recognize the device." The technical rate of the equipment is 100±10 punched cards per minute. The capacites of the feeding mechanism magazine and the receiving magazine are 700 and 600 punched cards respectively. The power supply is from a three-phase alternating current network with a voltage of 220/380 V (50 Hz). The power consumption is not more than 1.2 kV·A. The dimensions of the device are 1200 x 550 x x 1230 mm. Its mass is not more than 380 kg.

The device was developed by the Special Computer Design Office, Vil'nyus.

UDC 681.625

THE A-521-4 CHARACTER-SYNTHESIZING PRINTER



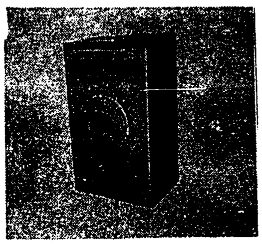
[Text] The device was created for the putput of alphanumeric and graphic information in M-6000/ /M-7000, SM-1 and SM-2 complexes. It has a sequential character-synthesizing impact principle of printing. The character is formed by a 5 x 7 mm matrix. The printing rate is at least 100 characters/s. The number of different printable characters is 96. There is a maximum of 120 characters per line and of two printed copies. The letter spacing is 2.34±0.3 mm. The character height is 2.54±0.5 mm and its width is 2±0.5 mm. The information carrier is paper with a width of 420 mm.* The space between lines is 4.23±0.5 mm. The time of movement along a line is not over 150 ms. The rate of graph mapping is at least 60 mm/s. The resolution in graph derivation is 0.423 mm. The device dimensions are 675 x 560 x 890 mm and its mass is not more than 65 kg.

The device is produced by the Orlovsk Control Computer Plant.

*Paper can be used which has a one-sided dye covering or is impregnated with a special chemical composition.

UDC 681.327.634

THE R-414-M SINGLE-DISK STORE WITH INTERCHANGEABLE RIGID MAGNETIC DISK

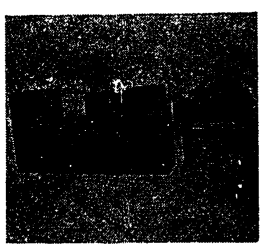


[Text] The device has been developed for the storage and reproduction of information in M-5100 and M-5010 computer complexes and the R-810 data preparation unit. It is used as a medium-capacity external memory with random access to information. It combines in a single device the functions of the R-412 and R-414 stores. For work in the regime of the R-414 store the rated capacities of the store and magnetic disk are 12.5 and 25 million bits and the rated information exchange rate is 1.6 million bits. When the R-412 is used as the store all the data can be reduced by half. Each disk surface has 128 tracks. The rotation frequency of the magnetic disk is 960 rpm. The mean track selection time is 135 ms. The longitudinal recording density on an internal track under R-412 and R-414 working conditions is 75 and 150 bits/minute respectively. The power supply is from a three-phase alternating current network with a voltage of 220 V (50 Hz). The device has dimensions of 1000 x 600 x 450 mm and a mass of not more than 140 kg.

The device is produced by the Panevezhis Precision Mechanics Plant.

UDC 681.321:681.3.06

THE "ISKRA-126," "ISKRA-1261" AND "ISKRA-1262" PROGRAM-CONTROLLED ELECTRONIC KEYBOARD COMPUTERS



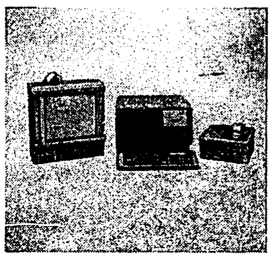
[Text] These are the first models of interactive multiprogram two-processor machines with built-in programming languages. They serve for the automation of routine elements of research, planning, design and engineering work in scientific research institutes and design offices; the solution of engineering and scientific research problems with direct participation of the user in the process of calculations and the formation of information on cathode-ray tubes, and also economic, planning and statistical calculation tasks; use as a means of collecting and processing information obtained in scientific research with a small number of sensors and the direct participation of the researcher in processing; work in subsystems for data collection, processing and registration on the basis of the International System of Small Computers and the Unified System of Electronic Computers; application as a programmed character-graphic terminal with the BASIC language.

Depending on the problems to be solved, the machines are supplemented by various sets of interface units and input-output devices and are issued in six models.

The devices were developed by the State Union Design and Technological Office for Computer Design, Leningrad.

UDC 681.324

THE "ISKRA-1256" COMPUTER CONPLEX



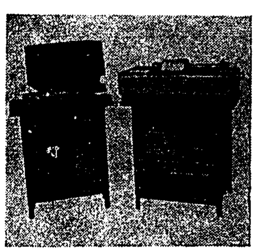
[Text] The device is intended for the collection and statistical processing of information in systems for the automation of scientific investigations with a small number of sensors and direct participation of researchers in processes of collection and processing of information (in particular, about a patient) arriving from medical instruments (or from a keyboard); the solution of engineering, scientific-technical, statistical reporting and economic problems of moderate complexity; the automation of routine elements of research, design and engineering work in the scientific research institute and the design office; work as a computer in a set of automated instruments, and also as a computer terminal.

The imput language is symbolic with bracket recording of arithmetic expressions. The program structure is operative with automatic numeration of line elements, has conditional and unconditional transitions to markers and subroutines, a cycle operator and switching of markers and subroutines. Digital variables have up to 12 decimal places. The variables have 256 characters. A mean time of 0.002 s is required for the operations of addition and subtraction, 0.006 s for multiplication and division and 0.2 s for elementary functions. Power consumption is 390 W. The device has dimensions of 500 x 525 x 335 mm.

The device is manufactured by the Kursk "Svetmash" Plant

UDC 681.171

THE "ISKRA-307" RAILROAD TICKET-CASH REGISTER



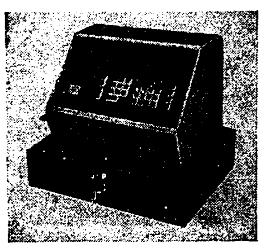
[Text] The machine was developed for the mechanization and recording of ticket and cash transactions in distant rail passenger traffic. Monetary receipts are recorded in a general summing register, cashier's receipts in a shift, the number of travel documents issued according to their categories, cost and the number of referenced tickets, the printing of travel tickets, bills of lading, checking tape, information output to a technical carrier, automatic determination of the cost of various types of travel tickets and data display in the process of formation of tickets on a cathode-ray tube.

Data input is from a mini-cassette and a keyboard. There are six adders (money registers). The money adders have a capacity of six decimal places and item counters a capacity of four decimal places. The capacities of the main and permanent memories are 4K x 8 and 8K x 8, that of a magnetic core storage is 1024 bits and that of a cathode-ray tube is 256 characters. The data output is indication on the cathode-ray tube, printing (of the travel document and the checking tape) and recording on punched tape.

The device was developed by the State Union Design and Technological Office for Computer Design, Leningrad.

UDC: 681.17:821.38

THE "ISKRA-363" ELECTRONIC MONITORING AND REGISTERING MACHINE



[Text] The machine was created for the automation of accounting and mechanization of the preparation of the corresponding documents, checking computational operations with residents and the performance of reporting operations on the rendering of services in hotels, hostels of the hotel type, etc.

The performed operations are: the input and the printing of conditionally permanent data, the input of the number of residents and the transitional balance, the type of payment, extra charges, correction for service according to the number of days and the type of service, return of the amount, printing of the daily report, readout and clearance.

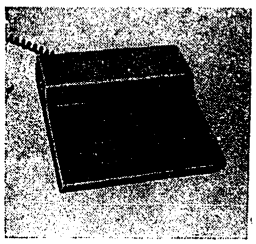
Data input into the machine is from a 10-key keyboard. Input data has up to seven decimal places. The main store data capacity is 256 x 4 bits and there are 19 registers. Data output is to a visual digital display and printer. Two documents (the travel document and a checking tape) are printed simultaneously. There are not more that 16 character positions on a line. The printing rate is 2.5 lines/s. Power consumption is 105 W.

The machine is produced by the Kursk "Smetmash" Plant.

UDC 681.321:621.38

THE "ISKRA-2240" KEYBOARD COMPUTER

[Text] The computer performs a wide range of economic planning, statistical reporting and accounting calculations: addition, subtraction, multiplication, division, square root extraction, change of sign, calculation of percentages, percentage ratios, discounts and surcharges, operations with a constant, exchange of contents of item registers, operations with a memory register (addition, subtraction, selection from a register with preservation of its contents, automatic



accumulation in memory registers of operands, results of operations and the number of times the "=" and "%" keys were pressed). The machine has 16 decimal places. Representation of the point is automatic with fixation and natural.

Three calculation regimes are accomplished: with and without rounding off and with rounding off with excess. The notation during input and output is decimal. Digital information is fed into the machine by means of a decimal keyboard and a point key, and operations are introduced by means of operating keys. Data output is on a visual display. Power consumption is not more than 10 W. The machine's dimensions are 250 x 240 x 55 mm. Its mass is not more than 3 kg.

The device is produced by the Kursk "Svetmash" Plant.

[23-2174]

2174
CSO: 1863

UDC 681.3

YES-1011 COMPUTING SYSTEM

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 8, 1980 pp 7-10

[Article by I. Kiss, Hungarian People's Republic]

[Text] The Hungarian company "Videoton," along with the production of the YeS-1010 computer, widely known in the Soviet Union, has starting producing the YeS-1011 computer, which is completely new in its manufacturing technology. It is a mini-computer, but thanks to its developed software it can be effectively used in areas characteristic of large-capacity computers. The machine is constructed on multi-layer printed circuit plates with the use of MOS/LS integrated circuits technology, which assures high speed with a low power consumption.

The large volume of a modular main store (a maximum of 1 Mbyte) permits simultaneously servicing several tasks independent of one another. The principle of multi-functional servicing is maintained by the main store protection system (segmentation), which excludes the possibility of loss or distortion of the data of separate problems.

Design execution of the YeS-1011 computer (Fig 1)

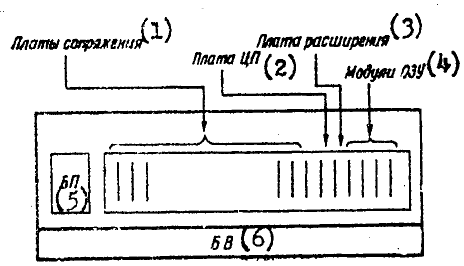


Figure 1. Design of the YeS-1011 system.

1 -- coupling plate
2 -- central processor plate
3 -- expansion plate
4 -- main store modulus
5 -- power unit
6 -- fan unit

The system is installed in a rack with the dimensions 537 x 483 x 377 mm, in which positions are allocated for the installation of one central processor plate (2), one expansion plate (3) to four main store modules (4), 14 coupling plates (1) with peripherals, a power unit (5) and a fan unit (6).

The minimum configuration includes one central processor plate, one expansion plate, one coupling plate for the connection of the operator's display, one main store module and one power unit.

An example of a possible configuration is shown on Fig 2.

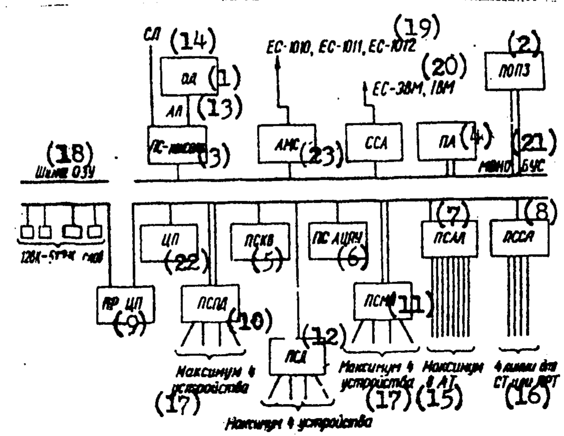


Fig 2.

1 -- operator's display console
2 -- operations plate with floating point
3 -- console-coupling plate-console
4 -- decimal arithmetic plate
5 -- card input coupling plate
6 -- alphanumeric printer coupling plate
7 -- asynchronous line coupling plate
8 -- synchronous line coupling plate
9 -- central processor expansion plate
10 -- package disk coupling plate
11 -- magnetic tape coupling plate
12 -- disk coupling plate
13 -- asynchronous line
14 -- synchronous line
15 -- maximum of 8 asynchronous terminals
16 -- 4 lines for synchronous terminals or RPT
17 -- maximum of 4 devices
18 -- main storage line
19 -- YeS-1010, YeS-1011 and YeS-1012
20 -- YeS-38M, YeS-18M
21 -- MONOBUS
22 -- central processor
23 -- intercomputer communications apparatus

Main central processor units of the YeS-1011

The microprogrammed memory is a device with a single-level principle of microprogramming in which the digits of a microinstruction directly control the apparatus. The volume of the microprogrammed memory is 2K words over 44 places with the possibility of expansion to 4K words. The time required for performance of one microinstruction is 250 nanoseconds.

The high-speed storage consists of 16 16-digit rapid registers (an instruction counter, a storage battery, base registers, an index register, etc).

The arithmetic logical device accomplishes 16 logical and 48 arithmetic operations on 16-digit operands.

The main units of the YeS-1011 computer also include indicators (there are four program and one microprogram indicators), a cyclical pulse generator, elements of input-output control (three universal instructions: SLO, AIO and HIO, a buffer to remember addresses or data, and one each devices for priority of access to the MONOBUS line, input-output control and interrupt processing).

Let us examine the functions of some units.

The central processor performs instructions on digits, bytes, words, double-length words and data chains of arbitrary length (a total of 141 computer instructions). The time is 1.9 microseconds for addition and 12 microseconds for addition with a floating point. The functions of the central processor also include addressing and control of main memory semicycles, diagnostic verification of main memory failures and the execution of instructions, control of interruptions (initiated by the equipment), and processing of program and equipment errors in the process of the work. The central processor also accomplishes the following addressing methods: DL--direct local; DG--direct global; IL--indirect local; EL--expanded local; ILX--indirect local with indexing; ELX--expanded local with indexing; IGX--indirect global with indexing; P--parametric; PX--parametric with indexing; RP--relative positive; RN--relative negative; IG'--indirect global special; IL'--indirect local special; EGX--expanded global with indexing.

It also accomplished integrated, microprogrammed functions: control panel functions, the starting and control of real-time clocks, the servicing of requests for programmed interruptions initiated by the operator or program user.

The YeS-1011 interrupt system is of the 64-level priority type, which is divided into two main groups: interruption requests initiated by orders of an operator or a user's program (levels 0-6); interruption requests initiated by the apparatus (by coupling devices upon the conclusion of input-output procedures or in the case of failure situations (levels 7-63). The microprogrammed exchange of context (the state of interrupting subroutines) is completed in 30 microseconds.

The priority of interruption requests of different levels depends on the location of the devices on the MONOBUS line.

Organization of input-output. The following devices are connected to the MONOBUS line: the central processor and an expansion plate, one to four main store modules with a maximum volume of up to 512K words; coupling devices which have direct access to the main store (of the DMA type), which are divided into mono- and multi-instructional (the former perform one channel instruction, the latter a series of instructions).

The SIO, HIO and AIO instructions are controlled by input-output procedures for DMA-type peripherals.

Program controlled coupling devices which have access to the main store only through the central processor can also be connected to the MONOBUS line. The input-output procedure is performed by the instructions WD and RD. For example, devices of a real time scale are connected directly to the MONOBUS line.

During the transmission of instructions and data through the MONOBUS line a 16-digit (Parallel) asynchronous method of signal exchange is used. The transmission rate is 2.3 Mbytes/second. Transmissions over the MONOBUS line and into the main store is controlled by the method of parity verification.

A master-slave relation is observed between connected devices, the "master" device being the controlling one in that case. Four types of transmission can be distinguished in that connection.

Master	Slave	Transmission type
Central processor	Main store	Data recording-readout
	Coupling device	Readout of interrupt code
Coupling device	Main store	Data recording-readout

The priorities system on the MONOBUS line. Priority between devices is observed during the processing of requests for access to the MONOBUS line and interruption requests.

Coupling devices and the central processor are connected with one another by means of a priority chain. Within the chain, priority is determined by the physical location of the given device on the MONOBUS line and is realized in the following manner: the central processor has the lowest priority and the coupling device, which is in the last position (the farthest from the central processor), the highest priority; between the two coupling devices one position can be allowed; the last position (a reference point) also can be empty, provided it is the next to last one occupied.

The central processor expansion plate accomplishes the following functions: addressing the maximum main store volume (up to 512K words), expansion of the microprogrammed store volume to 4K words, checking the parity of the transmitted information, and combination of the main store line with the MONOBUS line.

The main store and MONOBUS lines logically form a unified system.

The YeS-1011 main memory can be constructed on ferrite cores or metal-oxide semiconducting integrated microcircuits.

Technical characteristics of the main memory.

Ferrite main memories

Word length in digits	18 (16 information, 1 of parity, 1 of protection)
Time of complete cycle, ns	850
Volume in words	
one module	16K or 32K
maximum	128K (4 x 32K)

Semiconductor main memories

Word length in digits	18 (16 information, 1 of parity, 1 of protection)
Time of complete cycle, ns	600

Volume in words	
one module	32K, 64K, 128K
maximum	512K (4 x 128K)

Brief characterization of YeS-1011 system peripherals

The operator console display. The purpose of the coupling unit is to service one synchronous and one asynchronous communications line and control the real-time clock.

Since the YeS-1011 system does not have available an ordinary control panel (a total of three lamp indicators on the panel front), the performance of control panel functions is entrusted to the coupling plate and the operator console display.

The synchronous line has the following purposes: connection of a synchronous terminal or another computer, remote control of the system in two regimes: remote loading (loading of test programs or an operating system from the indicated peripheral) and a "text" regime (information input from a terminal into the main store of the system). The working rates are 1200, 2400, 4800, 9600 and 19,200 bauds.

The asynchronous line has the following purposes: connection of the operator console display (asynchronous terminal) which performs the control panel functions, assuring service and adjustment functions and constituting a means of communication between the user and the operating system. The working rates are 18.5, 27.5, 33.62, 36.5, 50, 75, 110, 134.49, 150, 200, 300, 600, 1200, 2400, 4800 and 9600 bauds.

The American Standard Code for Information Exchange (ASCII) is used (7 information digits and 1 parity digit).

The manufacturer plant recommends using the following devices as terminals:: VDS (+ matrix type DMA alphanumeric printer to order); VDT (+ matrix type DMA alphanumeric printer to order); VDT with two magnetic cassettes (+ any alphanumeric printer to order); VDDS (+ punched-tape input-output device and/or any alphanumeric printer to order); VDDS with two magnetic cassettes (+ any alphanumeric printer to order).

Also classed as peripherals are magnetic stores, the family of magnetic disks and the floppy magnetic disk. The coupling plate serves a maximum of four disks, in which case only one device can function at one time.

Technical characteristics of the carrier.

Number of tracks	77(0-76, 75th and 76th spares)
Access time, ms	83
Rotation frequency, rpm	360
Capacity in bytes:	
without format	388K
with format	255K
Transmission rate, bytes/s	20K

The carrier is compatible with the IBM 3741, IBM 3742, IBM 3747, BASF 6100 and SA 900. In the IBM format one track consists of 26 sectors, and a sector contains 128 bytes.

The IZOT 1370 magnetic disk. The coupling plate services a maximum of four devices, in which case one device consists of two disks.

Technical characteristics of a magnetic disk

Capacity in Mbytes:	
lower (fixed) plate	2.5
upper (interchangeable) plate	2.5
maximum	20 (4 x 5 Mbytes)
Mean store access time, ms	52.5
Rotation frequency, rpm	2400
Number:	
of surfaces	4
of cylinders	204
Information packing density, bits/mm	
on zero track	60
on 202nd track	90
Transmission rate, bytes/s	250K

Package disk with a 50-Mbyte capacity. The coupling plate services a maximum of four devices.

Technical characteristics of a package disk

Maximum capacity, in Mbytes	200(4 x 50 Mbytes)
Packing density, in bits/mm:	
on an internal track	87
on an external track	60
Number:	
of disks in a package	12
of useful surfaces	20
of tracks	406
of sectors	24
Rotation frequency, in rpm	2400
Positioning time, in ms:	
maximum	70
mean	35
Transmission rate, bytes/s	250K

A mini-disk with fixed heads has a capacity of 800K bytes. The coupling plate services a maximum of four devices.

Technical characteristics of a mini-disk

Maximum capacity, in Mbytes	3.2 (4 x 800 bytes)
Number:	
of disks	1
of useful surfaces	2

of tracks	256
of sectors	12
Rotation frequency, in rpm	3000
Mean store access time, in ms	10
Transmission rate, in bytes/s	153K

Let us note that all four types of disks can be serviced by the system simultaneously.

Magnetic tape stores. Used as stores are PT 305 and YeS-5017 devices. The coupling devices can service from one to four tape-drive mechanisms which have the following technical characteristics:

Number of tracks per magnetic tape	9
Packing density, bits/mm	32
Transmission rate, bytes/s	20K
Start-stop time, ms	14.4

Technical characteristics of the type VT 25153 alphanumeric printer

Line length	132 or 80
Set of characters	96 (64 in code ASCII + characters of Russian alphabet)
Printing rate, lines/min	650-1200
Paper advancement rate, mm/s	330
Character size, mm	2.4 x 1.65
Maximum line elevation time, ms	20
Number of copies	1-4
Inking tape width, mm	230
Inking tape length, m	27.5

Technical characteristics of the DZM 180 matrix alphanumeric printer

Line length, characters	132
Set of characters	96 (64 ASCII characters + characters of Russian alphabet
Printing rate, characters/s	180 (55-60 lines/min)
Character size, mm	2.54 x 2
Number of copies	1-4
Character depiction	5 x 7 point matrix

Let us note that the central processor services only one of the indicated printers in addition, they can be used for the output of infrom from the screen of a terminal.

The CR-601 card input. The coupling plate services one device. The input rate is 600 cards/minute. One card contains 80 (alphanumeric input) or 120 (binary input) characters.

Asynchronous lines. The coupling plate services a maximum of eight asynchronous (start-stop regime) terminals and controls the execution of a channel program by teletype, telegraph and telephone (V 2.4) interfaces, measage reception and transmission and special characters.

Technical characteristics of asynchronous lines

Transmission rate, bauds	75-9600
Character throughput of plate/sec	1000
Transmission type	Semiduplex, duplex
Maximum number of plates	16 (during expansion of the MONOBUS line)

The code system is International Telegraph Code No 5 or 2.

Proposed asynchronous terminals. Over teletype lines--teletypes (not issued by the plant); over telegraph lines--the VT 340 and the VDDS (both with telegraph interfaces); over telephone lines (V.24 interface)--the VT 340 (plus alphanumeric printer to order), the VDT (plus alphanumeric printer to order), the VDDS (plus alphanumeric printer to order), the VDDS (plus alphanumeric printer or punched-tape input-output device to order), the VDDS with two magnetic cassets (plus alphanumeric printer and two floppy disks to order).

Synchronous lines. The coupling plate services a maximum of four synchronous terminals.

Technical characteristics of synchronous lines

Transmission rate, bauds	1200-19,200
Plate throughput, characters/second	1500
Transmission type	Semiduplex, duplex
Maximum number of plates	16 (during expansion of the MONOBUS line)
Code system	ASCII, EBCDIC or binary

Proposed synchronous terminals: the VTS 56100, the RPT (Remote Process Terminal), the VDT, and the VDDS of synchronous regime.

The general-purpose coupling plate AMC (access mode channel) is intended for the creation of computer complexes based on computers of the given family (P-10, P-11 and P-12).

Technical characteristics of the AMC plate

Transmission method	Parallel (16-digit)
Transmission regimes	Simplex, semi-duplex, duplex
Maximum number of plates	8*
Throughput, words/s	Maximum of 600K
Maximum distance between two central processors of a set, m	7.5

*This assures the creation of 16 simplex, 8 semi-duplex and 4 duplex lines of communication.

The AMC coupling plate can also be used for the connection of peripherals (with a BSI interface) to the P-11 computer system.

The CCA (channel-channel adapter) plate serves for the creation of the following computer complexes: the YeS EVM P-11, the IBM P-11 and the Siemens P-11, in which the P-11 computer system is regarded on the part of a large-capacity computer as a peripheral working in a multiplex channel.

Technical characteristics of the CCA plate

Transmission method	Parallel (8-digit)
Throughput, bytes/s	15K - 20K
Maximum distance between the P-11 system and a large-capacity computer, m	60

The RPT controls processes in real time and services analog, binary and pulsed inputs and outputs.

Real-time devices connected directly to the MONOBUS line (Mini Benl-Time devices) provide 16 analog inputs, 2 binary inputs with two interruptions, 4 binary outputs and 8 multiplexed interruptions.

Let us note in conclusion that the P-11 (YeS-1011) computer system has great technical advantages of apossbilities over the P-10 (YeS-1011) and at the same time advantages with respect to software, which will substantially accelerate its introduction into industry.

[23-2174]

2174
CSO: 1863

UDC 681.325

PARALLEL COMPUTERS

Kiev IN-T ELEKTRODINAM. AN USSR. PREPR. [Institute of Electrodynamics, Ukrainian SSR Academy of Sciences: Preprint] in Russian No 223, 1980, 66 pages

KATKOV, A.F.

[From REFERATIVNYY ZHURNAL, AVTOMATIKA, TELEMEKHANIKA I VYCHISLITE'NAYA TEKHNIKA No 7, 1980 Abstract No 7B293 by T.M. Kuznetsova]

[Text] The author discusses fundamental aspects of the synthesis of special processors with maximum algorithmic productivity; the organization of the process of planning complicated systems in homogeneous computer systems with a rearrangeable structure; the principles of the construction of a device for processing commands in a high-productivity, parallel computer system; parallel, multiprocessor adaptive structure of data collection systems based on analog-to-digital channel processors. He suggests methods for transforming a virtual address in systems with segmented organization of the memory; microtests that do not require additional time, methods for the parallel stabilization of a single class of problems, and methods for the synthesis of plans for the parallel processing of information and the structure of multiprocessor computer complexes. He also analyzes the problems of MP [multi- or microprocessor] specialization for the solution of problems in systems for the digital, programmed control of machine tools, as well as the concept of an analog-to-digital, asynchronous, parallel computer system. The author also analyzes the processors of intercomputer interaction in distributed computer systems and the functional completeness of the exchange operations in homogeneous computer systems. He discusses the organization of programmed computer interaction in multicomputer complexes, parallel programming systems, and the automation of programming. He also investigates the problems involved in organizing parallel computations during the solution of several classes of algebraic problems and during the discrete modeling of physical processes. Finally, the author proposes an approach to parallel translation for multiprocessor structures with a parallel memory, as well as to the solution of counteremergency-control problems that is based on the parallel principles of information processing.

[22-11746]

UDC 681.325.5 181.48

ON SOME SPECIAL FEATURES OF THE MONITORING OF MICROPROCESSOR LARGE-SCALE INTEGRATED CIRCUITS

Moscow TR. IN-T ELEKTRON. UPR. MASHIN [Works of the Institute of Electronic Control Machines] in Russian No 76, 1979 pp 29-32

BEREZOV, Ye. P.

[From REFERATIVNYY ZHURNAL, AVTOMATIKA, TELEMEKHANIKA I VYCHISLITEL'NAYA TEKHNIKA No 7, 1980 Abstract No 7B320 by G.M. Kol'ner]

[Text] The author discusses the special features involved in monitoring microprocessor large-scale integrated circuits, using pseudorandom tests: pseudorandom codes are fed simultaneously into a standard large-scale integrated circuit and the one that is being checked, after which the outputs of these circuits are compared for coincidence. He mentions the necessity of including in the pseudorandom tests deterministic sequences that put the circuit being checked into a certain state. He also suggests that synchronizing control signals generated by the standard large-scale integrated circuit during the testing be used to control the monitoring process. References 1.

[22-11746]

UDC 772.99.002.5

A UNIT FOR MONITORING A LONG-TERM HOLOGRAPHIC MEMORY

Leningrad AVTOMATIZ. PROTSESSOV UPR. I OBRAB. INFORM. [Automation of Control and Information-Processing Processes] in Russian 1979 pp 91-95

VESELOV, I.M., GORLYANSKAYA, N.A., IL'INSKIY, V.P., MAKSIMOV, S.A., PIVOVAROV, V.T., and TSYPLYAYEV, S.I.

[From REFERATIVNYY ZHURNAL, AVTOMATIKA, TELEMEKHANIKA I VYCHISLITEL'NAYA TEKHNIKA No 7, 1980 Abstract No 7B305]

[Text] The authors discuss the structure and basic characteristics of a unit for the automatic monitoring of the brightness and location of the information points in the holograms in a long-term holographic memory. An additional uniform illumination channel that is frequency modulated has been introduced in order to improve the brightness measurement accuracy. In order to improve the accuracy of the monitoring of the points' locations, after preliminary guidance a tracking mode is realized for the light center of gravity of the point in the microhologram that is being investigated. In addition to this, there is suppression of the effect of non-linear distortions of the dissector's deflection yoke. The unit measures the brightness of light points with no worse than 5 percent accuracy and the displacement of their coordinates relative to their standard locations with an accuracy of no worse than 10 micrometers in the 16 x 16 mm field of the dissector's photocathode. Figures 1.

[22-11726]

UDC 681.325

AN AUTONOMOUS KAMAK-STANDARD SYSTEM BASED ON A MICROPROCESSOR

Novosibirsk IN-T AVTOMAT. I ELEKTROMETRII SO AN SSSR. PREPRINT [Institute of Automation and Electrometry, Siberian Department, USSR Academy of Sciences: Preprint] in Russian No 118, 1980, 37 pages

ZOLOTUKHIN, Yu.N., KRENDEL', Yu.M., YAKUSHEV, V.S., and YAN A.P.

[From REFERATIVNYY ZHURNAL, AVTOMATIKA, TELEMEKHANIKA I VYCHISLITEL'NAYA TEKHNIKA No 7, 1980 Abstract No 7B281 by T.M. Kuznetsov]

[Text] The authors describe an autonomous system that is constructed according to the principles of the KAMAK standard with the use of microprocessor equipment. The system consists of a microcomputer, a crate controller, a main memory with a capacity of 16 K byes, and an operator's console; they are joined together by means of a trunk communication link and connectors on their front panels. The autonomous system is based on a general-purpose, eight-bit microcomputer that is laid out on a single KAMAK card and is structurally formulated as a KAMAK module. The eight-bit microprocessor, which is realized on a single crystal and has a fixed architecture and system of commands, belongs to the 6800 family of microprocessors. It has three 16-bit and three 8-bit registers that are available for use by the programmer. The microcomputer's main memory is organized on the basis of large-scale integrated circuits of a static memory of the RAM type. The operator's console gives him the following capabilities: indexing and changing the contents of the processor's internal registers, running programs, indexing and changing the contents of the storage cells, stopping programs, step-by-step execution of programs, making entries in the memory with automatic increases in the successive addresses. Figures 22.

[22-11746]

SOFTWARE

UDC 681.142.2+518.5

ANALYSIS OF PROGRAM STRUCTURES

Kiev KIBERNETIKA in Russian No 1, 1980 manuscript received 17 Feb 78
pp 48-61

[Article by V. N. Kas'yanov]

[Text] The control flow in a program is usually shown in the form of a so-called control graph--an oriented graph whose peaks correspond to program operators and whose arcs reflect possible control transfers between them. In optimization of the translated program, one must solve a number of problems related to analysis of the structure of its control graph.

One of these problems is analysis of the program data flow to determine the properties of its states--those properties of data flow which are fulfilled at the moment it passes through one or another point of the program in any version of it. The conditions of the applicability of many optimizing transformations are formulated in terms of these properties. For example, the global economy of calculations [6], [25] requires that one find redundant expressions; the expression in some operator is redundant and can be economized if the current value of the expression has already been calculated regardless of the specific version of the program at the moment a given operator is carried out. The properties of program states are usually described [2, 19-21] by a set of invariant relations (with respect to a set of program versions) which relate its data to each other.

There are two of the most widely used approaches to finding these invariant relations. The first [1, 2, 6, 22], so-called interval analysis, is oriented toward representation of the control program graph in the form of a hierarchy of imbedded intervals--subgraphs of special type. The interval is characterized by the fact that it has a single peak for all arcs included in the interval, which lies on some closed path inside the interval. The second approach [18-20], the so-called iterative method, is related to consideration of a special type of numbering (or, which is the same thing, arrangements in linear sequences) of the control graph peaks, each of which gives an "intelligent" procedure, for the viewpoint of the effectiveness of the method, for processing the program operators within a single iteration.

Most optimizing transformations are related directly to specific structures of the control graph. For example, such transformations as a reduction of the force of operations or extension of invariant calculations [15, 16], are applied to the hierarchy of imbedded cyclically executed sections of the program. These "generalized cycles" in the control graph are said to be strongly bound regions or zones [15].

A significant time reduction of program execution is achieved by shifting the operators in the program to reduce the number of fulfillments of them. The effect of the indicated optimizing transformation increases if an entire fragment rather than a single operator is shifted. For example, transition from an Algol-program

```
for i: = 1 step 1 to 100 cycles
    begin entire x;
        x: = A[i];
        if x ≥ 0, then on M; x: = x ↑ 2;
        M: if B[i], then conclusion (x)
    end
```

to the program

```
for i: = 1 step 1 to 100 cycles
    if B[i], then begin entire x;
        x: = A[i];
        if x ≥ 0, then on M; x: = x ↑ 2;
        M: conclusion (x)
      end
```

cannot be accomplished by operator shifts. The shifted generalized operator in the control graph is called a hammock [5, 12]. The hammock is related to the remaining part of the control graph only by its initial and final peaks: all the arcs proceeding from the hammock are included in the final peak and all the arcs included in the hammock are included in the initial peak.

An important special case of the transformation described above is removal of operators from multiply repeated (executed) program sections (such as the zone or body of a recursive procedure), invariant with respect to different repetitions of the relieved section. The generalized operator removed from the repetitive section should be its linear component [10], [11], [13], [14]. The given type of hammock has the property that the relieved section is represented in the form of a multiply executed linear sequence of its own linear components. Let us explain the indicated transformation in the following simple example:

```
begin entire y, z;
    entire procedure A(k);
        begin entire x;
            x: = if y < 0, then z, otherwise 1;
```

```
                A: = if k = 0 then 1, otherwise x X A(k - 1)
            end;
        input (y, z); y: = A(10); output (y)
end
```

Here "x: = if y < 0 then z otherwise 1" is the invariant linear component of the body of recursive procedure A. The resulting program has the form:

```
begin entire x, y, z;
    entire procedure A(k);
        A: = if k = 0 then 1 otherwise x X A(k - 1);
    input (y, z); x: = if y < 0 then z otherwise 1;
    y: = A(10); output (y)
end
```

Representation of the control graph in the form of a hierarchy of imbedded hammocks is useful for many reasons [4, 5, 9-14, 23]. Specifically, it permits one to reduce the laboriousness of most global optimizations by factorized application of them. In factorization, each applicabion of the optimizing transformation touches on only one level of hammock imbedding, while all hammocks located at the deeper level appear in generalized form as inseparable structures.

Algorithms for separation of the linear components and hammocks are presented in the given article. The basis of the algorithms is four special indexes of the control graph peaks. These indexes can be used in the iterative method of analyzing data flow (see theorem 6). The hammock separation algorithm constructs a system of imbedded zones of the control graph during its operation (see theorem 3) and either represents it in the form of a hierarchy of imbedded intervals (if the interval representation of the given graph exists--theorem 5) or indicates the representation that does not exist (see theorem 4). The algorithm for separating the linear components simultaneously finds the bicomponents of the graph (see theorem 1).
The model of a computer with arbitrary
The model of a computer with arbitrary memory access is used when estimating the laboriousness of the algorithms. Formal description of this model can be found in [3]. It is assumed that any calculating operation or operation on control is carried out per unit time in the machine; all numbers should be whole numbers with an absolute value $O(n + m)$ if the graph being processed has n peaks and m arcs. The following notation is used here and below for specifying the boundaries: if f, g and h are functions of x, then the equality $f(x) = O(g(x) + h(x))$ denotes the existence of the constants k_1, k_2 and k_3 such that $|f(x)| \leq k_1|g(x)| + k_2 |h(x)| + k_3$ for all values of x.

The algorithm for separating the linear components has time complexity $O(m)$. This algorithm is optimum with accuracy up to the constant since each arc of the graph should be processed during separation of its linear components. The time complexity of the hammock separation algorithm comprises $O(nm)$.

This algorithm is not linear, but the number of arcs in the control graph of a real program is comparable to the number of peaks (usually $m \leq 2n$), while the number of hammocks in it will be on the order of n^2 at $m = n - 1$.

The article consists of five sections. Some concepts from graph theory and flow analysis theory of programs are given in the first two sections; indexing of the peaks of the control graph which are the basis of the considered algorithms are described. The next two sections are devoted to consideration of the properties of the described indexes and algorithms, including estimates of their laboriousness. In conclusion some comments are presented which touch on the practical use of the indicated algorithms.

Main Concepts

The following terms from graph theory are used below.

The oriented graph or graph G is an ordered pair (X, U), where X is a nonempty set of peaks and $U \subseteq X \times X$ is a set of arcs. If $(p, q) \in U$, it is said that p is a predecessor of peak q, while q is a successor of peak p. Let us denote the set of predecessors of peak p by PRED (p) and let us denote the set of its successors by SUCCESS (p). The graph is trivial if it consists of a single peak and contains no arcs.

The ordered sequence of peaks $(q_1, q_2, \ldots, q_k)$, $k \geq 1$, is the path from q_1 to q_k in graph G if for any $i \in [1: k - 1]$ we have $q_{i+1} \in$ SUCCESS (q_i). The sequential set of integers $\{k: i \leq k \leq j\}$ is denoted here and below by $[i : j]$. The path is assumed to be simple if it does not contain the same peak twice. The peak p is reachable from peak q in G if there exists a path from q to p in G. The graph in which any peak is reachable from each peak is said to be strongly connected.

The graph $G_1 = (X_1, U_1)$ is a subgraph of graph G if $X_1 \subseteq X$ and $U_1 = U \cap (X_1 \times X_1)$.

The strongly connected subgraph $G_1 = (X_1, U_1)$ of graph G will be a strongly connected component or bicomponent of graph G, which for any strongly connected subgraph $G_2 = (X_2, U_2)$ of graph G either $X_2 \subseteq X_1$ or the intersection $X_1 \cap X_2$ is empty.

Definition 1. A graph with two separated peaks--beginning and end--is called a control graph if each peak lies in the paths of the graph from the initial peak to the final peak, the initial peak has no predecessors and the final peak has no successors.

Let us subsequently denote the control graph with initial peak p_0 and final peak q_0 by G; G has $n = |X|$ peaks and $m = |U|$ arcs. It should be noted that $m \geq n - 1$. However, let us denote some subset of peaks of the graph and its subgraph for which the set of peaks is this subset.

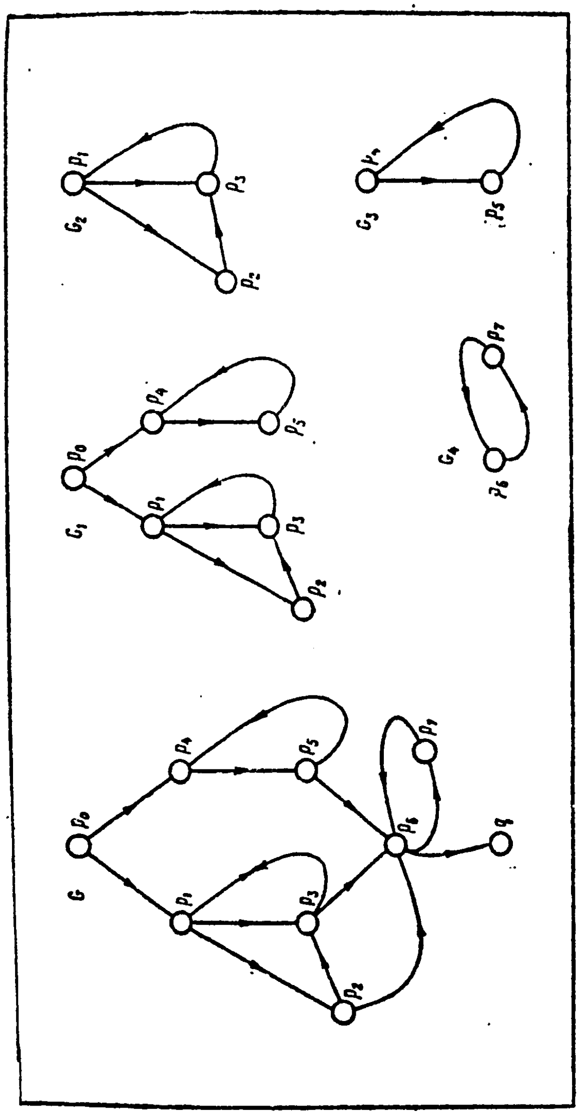


Figure 1. Example of Control Graph and Its Hammocks

Definition 2. A nontrivial strongly connected graph is a strongly connected region or zone [15].

Definition 3. Peak p of the subgraph is its initial peak in G if either $p = p_0$ or p has a predecessor in G not belonging to this subgraph. Peak p of the subgraph is its input peak in G if there exists a path from p_0 to p not containing the peaks of the subgraphs distinct from p. The external successor of the peaks of the subgraph is called its final peak.

Definition 4. The subgraph of graph G having a single initial and single final peak is a hammock [12] if this final peak is not a predecessor of the initial peak of the subgraph in graph G.

Definition 5. The subgraph of graph G having a single initial peak is called an interval [6] if this initial peak belongs to each zone of the given subgraph.

Definition 6. The hammock which satisfies the three following properties is called the linear component [12] of graph G:

1) both initial and final peaks of the hammock lie on each path from p_0 to q_0;

2) the initial peak of the hammock is not reachable from its final peak in G;

3) the given hammock is minimum with respect to the first two properties, i.e., no other hammock exists which satisfies the first two properties and which is contained in the given hammock.

Control graph G and all its hammocks is shown in Figure 1. Hammocks G_1 and G_4 are the linear components of graph G. We note that G contains three nontrivial bicomponents: $\{p_1, p_2, p_3\}$, $\{p_4, p_5\}$ and $\{p_6, p_7\}$.

Most algorithms for global analysis of data flows in programs are oriented by intervals with respect to the reduced control graphs, first introduced in [6]. The criterion of interval reducibility of the graph described below is taken from [7]. Let us say that graph $G_2 = (Z, V)$ is found from graph $G_1 = (X, W)$ by replacing some set $Y = \{Y_1, Y_2, \ldots, Y_k\}$ $(k \geq 0)$ of nonintersecting subgraphs of graph G_1 by peaks if , and

$(p, q) \in V$ if and only if either $\{p, q\} \subseteq Z \setminus Y$ and $(p, q) \in W$ or $\{p, q\} \cap \cap Y \neq \emptyset$, $p \neq q$ and $p \times q \cap W \neq \emptyset$.

Definition 7. Graph G is reducible (by intervals) if there exists the sequence of graphs $G_1 = G, G_2, \ldots, G_s$, called its zone-interval representation such that G_s contains only trivial bicomponents, while each graph G_i, $i \in [2: s]$ is found from graph G_{i-1} by replacing some set of nonintersecting strongly connected intervals by the peaks.

An example of zone-interval representation of a reduced graph is shown in Figure 2.

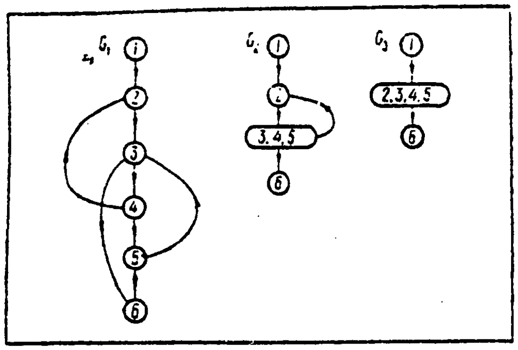


Figure 2. Zone-Interval Representation of Reduced Graph

We note that zone-interval representation of graph G can be used in all algorithms for global analysis of data flow in a program [20] which require its representation in the form of interval imbedding and also in all optimization algorithms [15] based on representation of the program in the form of a hierarchy of zones.

Definition 8. Let us call the set of zones D a hierarchy of zones of the graph [17] if the following two properties are fulfilled:

1) we have either $S_1 \subset S_2$ or $S_2 \subset S_1$ or the intersection S_1 and S_2 is empty for any two different zones S_1 and S_2 from D;

2) there exists the zone $S_1 \in D$ for any zone S of graph G such that $S \subseteq S_1$ and zones S and S_1 have a common input peak.

Indices of the Graph Peaks

The graph processing algorithms described in the paper are expressed in terms of a special type of indices of its peaks.

Definition 9. A one-to-one function which maps the set of its peaks X onto set $[1 : n]$ is an index of the peaks of control graph G.

If F is some index, then the F-number of peak p is denoted by F(p), while the set of sequentially F-indexed peaks $\{q \in X : F(q) \in [i:j]\}$ is denoted by $F[i:j]$ or by $F[u:v]$, where $F(u) = i$ and $F(v) = j$.

Definition 10. Let $(q_1, q_2, \ldots, q_k)$, $k \geq 1$, be the F-path if $F(q_i) < F(q_{i+1})$ is fulfilled for each $i \in [1:k-1]$. Peak p is F-reachable from peak q if there exists a F-path from q to p.

The class of "reasonable" procedures for processing the peaks of control graphs of programs is defined in [18].

Definition 11. The index F of graph G is reasonable if the two following properties are fulfilled:

1) for any two different peaks--p and q, if each path from p_0 to q contains p (i.e., p must precede peak q), then $F(p) < F(q)$;

2) if G is a reducible graph, then any simple path from p_0 will be a F-path (i.e., F controls the arrangement of graph G [12]).

Definition 12. Let $F(p) = i$. Then the F-region of peak p (let us denote it by F[p] or F[i]) will be the set of those peaks of graph G from which peak p in subgraph F [i:n] is reachable.

Before going to description of the algorithm for finding the F-region by its peak, it is appropriate to make some comments. Since we will be interested mainly only in the order of the time increase required to fulfill the algorithm in consideration of the algorithms, there is no need to describe the algorithms directly in terms of machine operations with arbitrary access. To make the description of the algorithms clearer, one can remove the easily realizable parts from it by using a language of higher level than the machine language. The time complexity of fulfilling each operator used below the language is not great and does not depend on the dimensions of the graph being processed; the programs represented below in this language can be translated directly to programs for a machine with arbitrary access to the memory. There are such traditional programming constructions in the language as variables, expressions, operators and procedures. Unlike existing programming languages, the elementary operators of the language under consideration are not only the operators of conferring or procedures, but also arbitrary verbal descriptions of the actions, the details of realization of which are obvious and insignificant. Another important difference is the presence of such types of data in the language as sets, mappings, flows and graphs.

The procedure which separates F[p] for given index F and given peak p is represented below (the notation close to ALGOL is used here an in other descriptions of the algorithms):

```
function REGION (F, p)
    include p in empty set A
    include p in empty set B
    until A is a nonempty cycle
        select and remove any q from A
        for all w from PRED (q) cycle
            if F(w) > F(p) and w ∉ B
            then add w to B
                 add w to A
```


APPROVED FOR RELEASE: 2007/02/08: CIA-RDP82-00850R000300070013-7

```
            all
         all
      all
      return B
   end
```

Since single processing of the graph arcs contained in the peaks of the separated F-region is accomplished upon fulfillment of the procedure, then the operating time of the algorithm can be estimated by O(k), where k is the number of these arcs.

Definition 13. The set of those peaks of the subgraph F[i:j] of which its output peaks are F-reachable in the subgraph, is called the F-line in some [i:j]. The F-rank of peak q in [i:j] is equal to max (i, {F(p):p ∈ A}), where A is a set of those peaks of the F-line in [i:j] of which peak q is F-reachable.

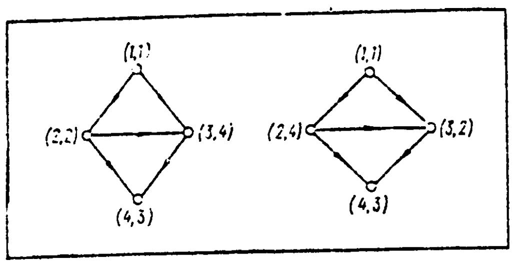


Figure 3. Graph and its Possible Indices N and M (of the two numbers which comprise each peak, the N-number is indicated by the first in the parentheses)

A procedure is presented below which determines R(p)--its F-rank in [i:j] by the given index F and by two numbers--i and j--for each peak p of F[i:j]:

```
function RANK (F, i, j)
   LINE is the set of all peaks of F[i:j] having external predecessors
   set R(v) equal to i for any peak v of R[i:j]
   for s from j to i through __-cycle
      select peak v for which F(v) is equal to s
      if v ∈ LINE
      then for all w of PRED (v) cycles
         if i ≤ F(w) < F(v)
         then add w to set LINE
         all
      all
   all
   for s from i to j through 1 cycle
      select peak v for which F(v) is equal to s
      if v ∈ LINE
```

```
            then set R(v) equal to F(v)
            all
            for all w from SUCCESS (v) cycles
                if F(v) < F(w) ≤ j
            then set R ≤ (w) equal to
                the maximum of R(v) and R(w)
                all
            all
        all
        return R
    end
```

It is easy to check that the procedure RANK has time complexity $O(m)$ since each arc of the graph is processed no more than three times, while each processing of the arc occupies a fixed time. It is assumed here and below that each index F is realized in the algorithms such that the time complexity both of finding its F-number by the given peak and of finding the peak by the given F-number is estimated by some constant.

Definition 14. Let us denote the index of the peaks of graph G by M at which we have:

1) $M(p_0) = 1$;

2) there exists the predecessor q such that $M(q) < M(p)$ for any peak p distinct from p_0;

3) for any pair of peaks--pa and q, if $(p, q) \in U$, $M(p) < M(q)$ and set $M[M(p) + 1: M(q) - 1]$ contains no predecessor of peak q, then each successor of the peaks of $M[M(p) + 1: M(q) - 1]$ has a M-number less than $M(q)$.

Definition 15. Let us denote the index of peaks of graph Q by N at which we have $N(p) < N(q)$ with respect to some index M for any pair of peaks--p and q--if and only if one of the following properties is fulfilled:

1) there exists a M-path from p to q;

2) $M(q) < M(p)$ and there does not exist a M-path from q to p.

We note that indices N and M always exist for an arbitrary graph G; they are not unique in the general case. Nevertheless the unique N-index corresponds to any M-index; the reverse statement is not true (Figure 3).

A procedure is described below which plots the indices N and M of the peaks of graph G:

```
    procedure INDEX 1
        N1: = n; M1: = 2
        set M(p0) equal to 1
```

```
    place p0 in empty FLOW
    until FLOW is not an empty cycle
        let q be the peak at the top of FLOW
        if the set SUCCESS (q) is empty
        then set N(q) equal to N1
            N1: = N1 - 1
            remove q from the top of FLOW
        otherwise select and remove some v from
            SUCCESS (q)
            if v has no M-number
            then place v in FLOW
                set M(v) equal to M1
                M1: = M1 + 1
            all
        all
    all
end
```

The numbers M and N are attributed to the peaks of graph G during inspection of all its peaks during backtracking through the graph, beginning from its initial peak. This process can be described as realized as a process of constructing some path P through G proceeding from p_0[8]. FLOW is used to store the given path. In this case the M-numbers indicate the order of including the peaks in path P, while the N-numbers indicate the order inverse to the order of removing the peaks from path G. Plotting of P begins by inclusion of peak p_0 in empty path P. Let some path p = = $(p_1, p_2, \ldots, p_k)$, $k \geq 0$, be plotted. If path P is empty (i.e., $k = 0$), the algorithm completes its work; otherwise path P changes by the following rules: if there exists a successor p of peak p_k which was earlier not included in P, the path is extended due to this successor: $P = (p_1, p_2, \ldots, \ldots, p_k, p_{k+1} = p)$. If this successor does not exist, the path is shortened due to removal of its last peak: $P = (p_1, p_2, \ldots, p_{k-1})$.

It is easy to see that the numbers of the peaks of the graph are assigned according to definitions 14 and 15. The algorithm occupies time O(m) since each peak of the graph is added to path P exactly once, while each arc proceeding from the peak of the path is inspected only once.

We note that there is a detour of the graph in constructing path P by a method similar to the Basis Index Algorithm [17] or to the method of depth first search [24] when all the peaks of the graph are inspected by alternation of transitions from the predecessor to the successor and vice versa. The N-index (similar to the Basis Index [17] and rENDORDER index [19] reflects the order inverse to that in which transition from peaks to their predecessors is accomplished). The M-numbers (similar to program numbers [24]) indicate the order in which the first transition to the peaks occurs.

Let us assume that some indices N and M of graph G connected to each other are fixed. Let us consider their M-numbers as the names of the peaks of the graph in all the given examples of graphs.

Definition 16. Let us call peak p a bipeak if we have $p \bar{\in} N[i]$ for any $i < N(p)$.

Definition 17. Let us denote the index of peaks of graph A by T at which the following properties are observed:

1) $T(p) < T(q)$ is fulfilled for any two bipeaks--p and q--if and only if $N(p) < N(q)$;

2) we have $N[p] = T[T(p): T(p) + |N[p]| - 1]$ for any bipeak p.

Let us consider the graph shown in Figure 4, a. There are three bipeaks: 1, 2 and 7 in this graph. In this case only the N-region of bipeak 2 is not a trivial subgraph (it is formed by the set of peaks {2, 3, 4, 5, 6}). The graph contains two linear components: {1} and {2, 3, 4, 5, 6}, each of which has the form $T[T(p): T(q) - 1]$, where p and q are the initial and final peaks, respectively, of the indicated linear component. We note that each bicomponent of the graph coincides with N[p] for some bipeak p.

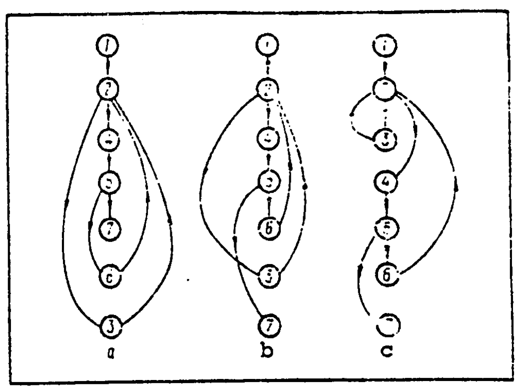


Figure 4. The Arrangement of the Peaks of the Graph Also Reflects Its Possible Indices: N(a), K(b) and L(c)

One of the possible realizations of the algorithm which accomplishes T-indexing of the peaks of graph G is represented below. The procedure is perceived as its own input of the N-number of the peaks of the graph, uses the procedure REGION to find the N-regions of the bipeaks, assigns the T-numbers to the peaks and finds the set of BIPEAKS of the graph.

```
procedure INDEX 2 (N)
    let the BIPEAKS be an empty set
    Tl: = 1
    for k from 1 to n through 1 cycle
        select peak z at which N(z) is equal to k
```

```
            if z has no  T-number
            then set T(z) equal to T1
                T1: = T1 + 1
                add z to set BIPEAKS
                assume set A is equal to REGION
                (N, z) \ {z}
                until A is not an empty cycle
                    select and remove some v of A
                    set T(v) equal to T1
                    T1: = T1 + 1
                all
            all
        all
    end
```

Each peak of the graph belongs to one separated N-region inaccuracy. Thus, the algorithm occupies time $O(m)$ since each arc is inspected only once upon separation of the N-region containing the peak which includes this arc.

Definition 18. Index K is determined as the last term of the sequence $N_1 = N, N_2, \ldots, N_n = K$, in which we have the following for any $i \in [1:n - 1]$.

1. $N_{i+1}(p) = N_i(p)$ for each $p \in N_i[1:i - 1]$.

2. For any two peaks--p, $q \in N_i[i: n]$, $N_{i+1}(p) < N_{i+1}(q)$ is valid if and only if one of the properties is fulfilled:

   a) $p \in N_i[i]$ and $q \bar{\in} N_i[i]$;

   b) $N_i(p) < N_i(q)$ and either $\{p, q\} \subseteq N_i[i]$ or $p \bar{\in} N_i[i]$ and $q \bar{\subset} N_i[i]$.

Let us consider once more the graph shown in Figure 4. Here $N_1 = N_2 = N$, $N_3 = N_4 = N_5 = N_6 = N_7 = K$.

A procedure is described below which perceives N_i-numbers of the peaks of the graph as the input and which issues their N_{i+1}-numbers by variation of some of the written numbers:

```
    procedure RENUMBERING 1 (F, i)
        select the peak z at which F(z) is equal to i
        make the set A equal to REGION (F, z)
        number 1: = 1; number 2: = i + |A|
        for k from 1 to n through 1 cycle
            select peak u at which F(u) is equal to k
            if k < i
            then set F1(u) equal to k
            otherwise if u ∈ A
                then set F1(u) equal to number 1
                    number 1: = number 1 + 1
                otherwise set F1(u) equal to number 2
```

```
                number 2: = number 2 + 1
            all
        all
    all
    take F1 as F
end
```

It is easy to see that O(m) is the time complexity of the described procedure and consequently the total working time of the K-index algorithm can be estimated as O(nm).

Definition 19. Index L is determined as the last term of the sequence $L_n = K, L_{n-1}, \ldots, L_0 = L$ in which we have the following for each $i \in [1:n]$.

1. $L_{i-1}(p) = L_i(p)$ for any p not contained in $L_i[i:i + |L_i[i]| - 1]$.

2. For any two peaks--p and q, there will be $L_{i-1}(p) < L_{i-1}(q)$ from $L_i[i: i + |L_i[i]| - 1]$ if and only if one of the properties is fulfilled:

   a) p and q have the same L_i-rank in $[i: i + |L_i[i]| - 1]$ and $L_i(p) < L_i(q)$;

   b) the L_i-rank of peak q in $[i: i + |L_i[i]| - 1]$ exceeds the corresponding L_i-rank of peak p.

A graph is shown in Figure 4 where $L_7 = L_6 = L_5 = L_4 = L_3 = L_2 = K$, $L_1 = L_0 = L$, $L_2\ [L_2(2)] = \{2, 3, 4, 5, 6\}$ and consequently $L_2[2:6] = \{2, 3, 4, 5, 6\}$. Here the L_2-line in [2:6] is formed by peaks 2, 4 and 5. Peaks 2 and 3 have the same L_2-rank in [2:6] the same as peaks 5 and 6. Thus, the graph contains four nontrivial hammocks: {1, 2, 3, 4, 5, 6}, {2, 3}, {2, 3, 4} and {2, 3, 4, 5, 6}, each of which consists of sequentially L-numbered peaks.

Another example of possible L-index of its peaks is presented in Figure 5. Here the peaks of the graph are noted by M-numbers; the numbers N and L of the peaks of the graph are indicated in parentheses (the first is the number N) alongside the corresponding peaks.

A procedure is described below which accomplishes for any $i \in [1:n]$ transition from L_i-index of the peaks of the graph to L_{i-1}-index:

```
procedure RENUMBERING 2 (F, i)
    select the peak u at which F(u) is equal to i
    j: = i + |REGION (F, u)| - 1
    set R equal to RANK (F, i, j)
    set NUMBER (s) equal to zero for each s from [i: j + 1]
    for all v from F [i:j] cycle
        set NUMBER (R(v) + 1) equal to NUMBER (R(v) + 1) + 1
    all
```

```
        for s from i to j through 1 cycle
            set NUMBER (s + 1) equal to NUMBER (s + 1) + NUMBER (s)
        all
        for s from 1 to n through 1 cycle
            select peak v at which F(v) is equal to s
            if s ∈ [i:j]
            then the note at the given moment for any t ∈ [i:j]
                NUMBER (t) denotes the number of the peaks v from F[i:j]
                at which either R(v) < t or R(v) = t and F(v) < s end of
                note
                set F1(v) equal to i + NUMBER (R(v))
                set NUMBER (R(v)) equal to NUMBER (R(v)) + 1
            otherwise set F1(v) equal to F(v)
            all
        all
        take F1 as F
    end
```

Calculation of j and R occupies time O(m) and O(n) is the time required to rewrite the F-numbers. Thus, O(m) is the laboriousness of the procedure RENUMBERING 2 and consequently the total working time of the algorithm for accomplishing the L-index of the peaks of the graph can be estimated as O(nm).

Finding the Linear Components

Fulfillment of the algorithm, which finds the linear components of the graph, consists of three steps. The N-index of the peaks of the graph is established in the first step. The second step precedes the index obtained in the first step as the input, establishes the T-index of the peaks of the graph and finds the set of its BIPEAKS. Essentially, the linear components are separated in the third step when the peaks of the graph are processed in the order of increase of their T-numbers. This part of the algorithm relies on the following properties of the previously introduced indices of the peaks of the graph.

Lemma 1. Let $P = (p_1, \ldots, p_s)$ be the simple path through graph G such that $M(p_1) = \min\{M(p_j) : j \in [1:s]\}$. There then exists a M-path from p_1 to p_j for any $j \in [1:s]$.

Proof. Let us carry out the proof by induction with respect to s. The result is trivial for $s = 1$. Let us assume that the lemma is valid for all paths P consisting of less than s peaks and let us consider the path consisting of s peaks.

It is easy to see that either $M(p_s) > M(p_{s-1})$ or $M(p_s) < M(p_{s-1})$. For the first case the result follows directly from definition 10 and the inductive premise.

Let $M(p_s) < M(p_{s-1})$. This means that $M(p_1) < M(p_s) < M(p_{s-1})$ and consequently there exists $j \in [1:s - 1]$ such that $M(p_j) < M(p_s) < M(p_{j+1})$. We find from definition 14 that for any three peaks--q_1, q_2 and q_3--such that $(q_1, q_2) \in U$ and $M(q_1) < M(q_3) < M(q_2)$, there exists the predecessor q of peak q_3 for which $M(q_1) \leq M(q) < M(q_3)$. Multiple use of this remark (beginning at peaks p_j, p_{j+1} and p_s) in view of the finiteness of the graph leads to finding the M-path from p_j to p_s. The presence of this path, according to the inductive premise, completes the proof. The lemma is proved.

Corollary 1. There exists a M-path from the initial peak of the graph to any peak $p \in X$. Hence, in view of definitions 12 and 15, we find the following properties.

Corollary 2. $N(p_0) = 1$ and there exists the N-path from peak p_0 to any peak $p \in X$.

Corollary 3. The following properties are equivalent for any peak $p \in X$ and for any peak q from N[p]:

1) q is the input peak of subgraph N[p];

2) q is the initial peak of subgraph N[p];

3) there exists the predecessor v of peak q for which $N(v) < N(p)$.

Lemma 2. Subgraph N[i] is strongly connected for any $i \in [1:n]$.

Proof. Let N[i] not be a trivial subgraph and p is its peak whose N-number is equal to i. Let us consider peak q of N[p] distinct from p and let us assume that $M(q) > M(p)$. We then find from definitions 12 and 15 the existence of the M-path from p to q. According to definition 15, this indicates the presence of a N-path from p to q which belongs to subgraph N[i], according to definition 12. At the same time the presence of a path from q to p in subgraph N[i] also follows from definition 12.

Thus, to prove the lemma it is sufficient to show that $M(p) < M(q)$. Let us assume that this is not so and let us consider the path $P = (p_1 = q, p_2, \ldots, \ldots, p_3 = p)$ in N[i:n]. It exists according to definition 12. According to our premise, $M(p) > M(q) \geq M(p_j) = \min\{M(p_k): k \in [1:s]\}$ and consequently in view of lemma 1 there exists the M-path from p_j to p where $p_j \in N[p]$ and $p_j \neq p$. According to definition 15 this means that $N(p_j) < N(p)$. Thus, a contradiction occurs. The lemma is proved.

Corollary 4. For any $p \in X$ we have $M(p) = \min\{N(q): q \in N[p]\}$ and each peak of N[p] is M-reachable from peak p.

Corollary 5. For any two different peaks of the graph--p and q--we find either $N[p] \subset N[q]$ or $N[q] \subset N[p]$ or the intersection $N[p] \cap N[q]$ is empty.

Theorem 1. Subgraph S is a bicomponent of graph G if and only if there exists the bipeak p for which $S = N[p]$.

Proof. Let S be the bicomponent of graph G and let p be the peak of S such that $N(p) = \min \{N(q):q \in S\}$. According to definition 12, $S \subseteq N[p]$. However, according to lemma 2 and corollary 5, $N[p] \subseteq S$ and p is a bipeak.

Let p be some bipeak. According to lemma 2, some bicomponent of the graph (let us say S) contains all peaks from N[p]. Let $S \setminus N[p]$ not be empty. There should then exist a peak q such that $N(a) < N(p)$ and q is reachable from peak p in $N[N(q):n]$. This means that $p \in N[q]$ and thus a contradiction occurs.

The theorem is proved.

Definition 20. Bipeak p is a concatenation point if T(p) coincides with the maximum T-number of the successors of peaks from $T[1: [(p) - 1]$.

Theorem 2. Let $\{p_1, \ldots, p_{k+1}\}$ be a set of all concatenation points of graph G arranged in the order of increase of their T-numbers. The subgraph H of graph G will then be a linear component with initial peak p and final peak q if and only if for some $j \in [1:k]$ we have $H = T[T(p_j): T(p_{j+1}) - 1]$, $p = p_j$ and $q = p_{j+1}$.

Proof. It is obvious that the theorem is valid if the graph contains only trivial bicomponents since in this case, according to theorem 1 and definition 17, we have $T(p) < T(q)$ for any arc of the graph (p, q).

Let us consider the general case. We note that if some peak of the bicomponent of the graph belongs to the linear component H, then H contains all the peaks of this bicomponent. Further, if p is either the initial or final peak of the linear component and belongs to some bicomponent of the graph, then p is the only input peak of this component. Let graph G_1 be found from graph G by replacing it by components by peaks. It follows from the properties described above that subgraph H of graph G is its linear component with initial peak p and final peak q if and only if there exists the sequence of linear components $H_1, H_2, \ldots, H_s$ of graph G_1 for which the following properties are fulfilled. Subgraph $\cup\{H_i: i \in [1:s]\}$ of graph G_1 corresponds to subgraph H of graph G; the initial peak for H_{i+1} for each i of $[1:s - 1]$ is the final peak for H_i corresponding to the bicomponent of graph G which has more than one input peak; the initial peak for H_1 and the final peak for H_s correspond to single-input bicomponents S_1 and S_2 of graph G, while p and q are the input peaks of S_1 and S_2, respectively. Thus, according to definition 6, corollaries 2 and 3 and theorem 1, the general case reduces to the case when G contains only trivial bicomponents.

The theorem is proved.

The third part of the algorithm for finding the linear components in the graph has the following form:

```
procedure LINEAR COMPONENTS (T, BIPEAKS)
    take peak p0 as u
    number: = 1
    for k from 1 to n - 1 through 1 cycle
        select the peak v for which T(v) is equal to k
        for all w from SUCCESS (v) cycle
            if the number < T(w) then the number: = T(w) all
        all
        select the peak z for which the peak T(z) is equal to k + 1
        if number = k + 1 and z ∈ BIPEAKS
        then T[T(u):T(z) - 1] is the linear component
            with initial peak u and final peak z
            take peak z as u
        all
    all
end
```

The given part of the algorithm calculates the maximum T-number of successors $T[1:T(p) - 1]$ for each peak p upon processing of the peaks of the graph in the order of increase of their T-numbers and correctly separates the linear components of the graph according to theorem 2. This part of the algorithm also requires time $O(m)$ since each arc of the graph is processed exactly once. Thus, the entire algorithm for finding the linear components occupies $O(m)$ time.

Finding the Hammocks

The algorithm of intersection of all hammocks of the control graph consists of four steps. Indices N and M of its peaks are established in the first step. The second step perceives the N-numbers of the peaks as the input and establishes the K-index by varying some numbers. The next step of the algorithm is related to introduction of the L-index. It uses K-numbers found in the previous step and accomplishes the required rewriting of them. The three indicated steps were described previously and occupy time $O(nm)$. Essentially, the hammocks are separated in the fourth step of the algorithm and utilize the M-index found in the first step and the L-index found in the third step. This part of the algorithm is based on the properties of indices of the peaks of the graph given below.

Lemma 3. For any $i \in [1:n]$ and any peak $p \in X$, there will be $K_i[p] = N[p]$.

Proof. Let us carry out the proof by induction with respect to i. The statement is obvious at $i = 1$. Let us assume that the lemma is valid for all $i < s$ and let us consider the case when $i = s$. If p is some peak of the graph, then either $K_{s-1}(p) < s - 1$ or $K_{s-1}(p) \geq s - 1$.

Let $K_{s-1}(p) < s - 1$. From definition 18 we find $K_{s-1}[K_{s-1}(p): n] = \bar{K}_s[K_s(p):n]$. This means that $K_s[s] = K_{s-1}[p]$ in view of definition 12 and consequently according to the inductive premise $K_s[p] = N[p]$.

Let $K_{s-1}(p) \geq s - 1$. It is easy to see that either $p \bar{\in} K_{s-1}[s - 1]$ or $p \in K_{s-1}[s - 1]$. Let us consider the first case. Here the intersection $K_{s-1}[s - 1] \cap K_{s-1}[p]$ is empty in view of the inductive premise and corollary 5. Thus, according to definition 18, $K_{s-1}[p] \subseteq K_s[K_s(p):n] \subseteq$ $\subseteq K_{s-1}[K_{s-1}(p):n]$ and consequently $K_s[p] = K_{s-1}[p] = N[p]$. Let us then consider the case when $p \in K_{s-1}[s - 1]$. From definition 18 we find that $K_{s-1}[K_{s-1}(p): n] \subseteq K_s[K_s(p): n]$ and if peak p is reachable from some peak q in $K_s[K_s(p): n]$, then $q \in K_{s-1}[K_{s-1}(p): n]$. Thus, even in this case $K_s[p] = K_{s-1}[p] = N[p]$. The lemma is proved.

Corollary 6. The set $\{K_i[i]: i \in [1: n]\}$ coincides with the set $\{N[p]: p \in X\}$.

Let us denote by D a set of all nontrivial subgraphs belonging to the set $\{K_i[i]: i \in [1: n]\}$.

Theorem 3. D is the hierarchy of the zones of graph G.

Proof. For any zone S if peak p is such that $N(p) = \min \{N(q): q \in S\}$, then $S \subseteq N[p]$. Hence, according to lemma 2 and according to corollaries 2, 3, 5 and 6, we find that D is the hierarchy of zones of graph G.

The theorem is proved.

Theorem 4. Let us reduce graph G if and only if each zone S of D has only a single initial peak.

Proof. It is shown in [7] that G is a reducible graph if and only if each of its zones has only a single input peak. This completes the proof according to theorem 3 and corollary 3.

The theorem is proved.

Let us determine the sequence of sets $D_0, D_1, \ldots, D_s$ in the following manner:

1) D_0 is an empty set;

2) let the sets $D_0, \ldots, D_{i-1}$ be constructed. Then if set $A = D \setminus \bigcup \{D_j: j \in [0: i - 1]\}$ is not empty, then D_i is formed of all minimum subgraphs of set A (i.e., of all subgraphs $S \in A$, which for any $S_1 \in A$ either $S \subseteq S_1$ or $S_1 \cap S$ is empty), and in the opposite case, i.e., when set A is empty, s is equal to i - 1.

Theorem 5. If G is a reducible graph, then $G_1, G_2, \ldots, G_{s+1}$ is its zone-integral representation where for any $i \in [0: s]$ graph G_{i+1} is found from graph G by replacing the set of its subgraphs D_i by peaks.

Proof. It follows from theorems 3 and 4 that for any $i \in [0: s]$ graph G_{i+1} is found from graph G_i by replacing the set of nonintersecting zones, each of which has only a single initial peak, by peaks. Thus, it remained to prove that any path $P = (q_1, q_2, \ldots, q_k = q_1)$ in each $S \in D$ either contains the initial peak of zone S or is entirely contained in some zone $S_1 \in D$ such that $S_1 \subset S$. To prove this, let us consider the peaks $q \in S$ and $q_i \in P$ such that $N(q_i) = \min \{N(q_j): j \in [1: k]\}$ and $N(q) = \min \{N(v): :v \in S\}$. We find from definition 12 and corollary 5 that either $q = q_i$ or $q \neq q_i$ and $p \subseteq N[q_i]$. This completes the proof according to corollaries 2, 3 and 6.

The theorem is proved.

Lemma 4. For any $i \in [0: n]$ and any peak $p \in X$, if $L_i(p) \leq i$, then $L_i[p] = N[p] = L_i[L_i(p): L_i(p) + |L_i[p]| - 1]$.

Proof. Let us carry out the proof by induction with respect to i. The result for $i = n$ follows directly from definitions 18 and 19, according to lemma 3 and corollary 5. Let us assume that the lemma is valid for all $i > k$ and let us consider the case when $i = k$. Let p be the peak of the graph such that $L_k(p) = j \leq k$. Then according to definition 19 we find $L_{k+1}(p) = j$ and $L_k[j: n] = L_{k+1}[j: n]$. This means that $L_k[j] = L_{k+1}[j]$. However, since $L_k(p) = j \leq k$, according to the inductive premise and corollary 5, we find $L_{k+1}[j] = L_{k+1}[j]: j + |L_{k+1}[p]| - 1] = N[p]$ and either $L_{k+1}[k + 1] \subset L_{k+1}[j]$ or $L_{k+1}[k + 1] \cap L_{k+1}[j]$ is empty and, consequently, according to definition 19, $L_{k+1}[j] = L_k[j: j + |L_{k+1}[p]| - 1] = N[p]$. This completes the proof since, as proved above, $L_k[j] = L_{k+1}[j]$. The lemma is proved.

Lemma 5. For any arc (p, q) of graph G, the following properties are equivalent:

1) $N(p) < N(q)$;

2) $K_i(p) < K_i(q)$ for any $i \in [1: n]$;

3) $L_i(p) < L_i(q)$ for any $i \in [0: n]$.

Proof. Let (p, q) be some arc of graph G. Let us prove by induction with respect to i that $N(p) < N(q)$ if and only if $K_i(p) < K_i(q)$. This statement is trivial for $i = 1$. Let us assume that the result is valid for all $i < s$ and let us consider the case when $i = s$. It follows from definition 18 that the ratio between K_{s-1} can differ by the numbers of peaks p and q from the ratio between their K_s-numbers if and only if either $p \in K_{s-1}[s - 1]$ and $q \in K_{s-1}[s - 1: n] \setminus K_{s-1}[s - 1]$ or $p \in K_{s-1}[s - 1: n] \setminus K_{s-1}[s - 1]$ and $q \in K_{s-1}[s - 1]$. But if $q \in K_{s-1}[s - 1]$ and $K_{s-1}(p) \geq s - 1$, then p also belongs to $K_{s-1}[s - 1]$ since (p, q) is the arc of graph G. Thus, it is sufficient to consider the case when $p \in K_{s-1}[s - 1]$ and $q \in K_{s-1}[s - 1: n] \setminus K_{s-1}[s - 1]$. Here, in view of definition 18, we have $K_s(p) < K_s(q)$. Let us assume that $N(p) > N(q)$. This means that $p \in N[q]$ and consequently, according to lemma 3, $p \in K_s[q]$, i.e., $K_s(p) > K_s(q)$. Thus, the given premise leads to a contradiction.

Now let us prove by induction with respect to i that $N(p) < N(q)$ if and only if $L_i(p) < L_i(q)$. As shown above, this statement is fulfilled if $i = n$. Let us make the inductive premise and let us consider the case when $i = s$. Upon proof of the inductive transition, it is sufficient to prove the impossibility of the situation when $L_{s+1}(p) > L_{s+1}(q)$ and L_{s+1}-rank of peak q exceeds the L_{s+1}-rank of peak p since other situations (see definition 19) related to variation of the ratio between the numbers of peaks p and q are clearly not possible upon transition from L_{s+1} to L_s. Let us assume that this situation is possible. Then according to definition 19 we have $L_s(p) < L_s(q)$ and according to the inductive premise, $N(p) > N(q)$. From $N(p) > N(q)$, we find $p \in N[q]$ and consequently, according to corollary 4, there exists the N-path from q to p. Thus, according to the inductive premise, there exists the L_{s+1}-path from peak q to p and the L_{s+1}-rank of peak q cannot be greater than the L_{s+1}-rank of peak p, which contradicts the premise. The lemma is proved.

Theorem 6. For any $j \in [0: n]$ the indices K_j and L_j are reasonable.

Proof. According to definition 11, lemma 5 and corollary 2, it is sufficient to prove that any simple path $P = (q_1 = p_0, q_2, \ldots, q_s)$ in reducible graph G is an N-path. Let us assume the opposite, i.e., let us assume that for some simple path P in reducible graph G there exist those $i \in [1: s - 1]$ such that $N(q_i) > N(q_{i+1})$ and let us consider the least value of i. Then $q_i \in N[q_{i+1}]$ and there exists the path from p_0 to q_i which does not pass through peak q_{i+1}. Thus, according to corollaries 2 and 3, subgraph $N[q_{i+1}]$ has more than one initial peak, which contradicts the reducibility of the graph according to theorem 4.

The theorem is proved.

We shall subsequently denote the hammock with initial peak p and final peak q by H and the set $H \cup \{q\}$ by $\bar{H}$.

Lemma 6. For any $i \in [1: n]$, either $N[i] \subseteq H$ or $\bar{H} \subset N[i]$ or the intersection $H \cap N[i]$ is empty.

Proof. It is easy to check that if S is some zone of graph G, then either $S \subseteq H$ or $\bar{H} \subset S$ or the intersection $S \cap H$ is empty. Hence, according to lemma 2, we find the necessary confirmation since, according to definition 4 and corollary 2, $N(p) = \min \{N(p_1): p_1 \in H\}$. The lemma is proved.

Lemma 7. Let $P_1 = (p_1, \ldots, p_k)$ and $P_2 = (q_1, \ldots, q_s)$ be the M-paths such that $M(p_1) \leq M(q_1) \leq M(p_k) \leq M(q_s)$. There then exists the M-path from q_1 to p_k.

Proof. Let us assume that there does not exist the M-path from peak q_1 to peak p_k. Then $M(p_1) < M(q_1) < M(p_k)$. We find from corollary 1 that set A of all peaks v such that $M(q_1) < M(v)$ is not empty and v belongs to M-paths from p_0 to peaks from P_1. Let u and z be the peaks of the graph such that $M(u) = \min \{M(v): v \in A\}$ and $M(z) = \max \{M(v): M(v) < M(u)$ and $v \in \mathrm{PRED}\ (u)\}$. Then $M(z) < M(q_1) < M(u)$, $z \in \mathrm{PRED}\ (u)$ and the set $B = M[M(z) + 1: M(u) - 1]$ does not contain predecessors of peak u. Hence, according to definition 14, we find that all the successors of the peaks from B have M-numbers less than the M-number of peak u. But, at the same time $q_1 \in B$, $M(u) < M(q_s)$ and P_2 is the M-path from q_1 to q_s. Consequently, some successor of the peaks from B has a M-number not less than M(u). Thus, a contradiction occurs. The lemma is proved.

Lemma 8. If $M(p) < M(q)$, then $N[N(p : N(q) \subseteq \bar{H} \subseteq N[N(p): n]$.

Proof. We find from definition 4 and corollary 1 that $M(p) = \min \{M(u): : u \in H\}$. According to lemma 1 and definition 15, this means that $\bar{H} \subseteq \subseteq N[N(p): n]$. Thus, to prove the lemma, it remained to show that $N[N(p): : N(q)] \subseteq \bar{H}$. In view of definition 15, we find $p_1 \in N[N(p) + 1: N[q) - 1]$ and $M(p) < M(q)$ if and only if one of the following properties is fulfilled:

1) $M(p) < M(P_1) < M(q)$ and there are M-paths from p to p_1 and from p_1 to q;

2) $M(p_1) < M(p) < M(q)$ and there is the M-path from p_1 to q and there is no path from p_1 to p;

3) $M(p) < M(q) < M(p_1)$ and there is the M-path from p to p_1 and there is no M-path from q to p_1.

If properties 1 or 3 are fulfilled, in view of definition 4, $p_1 \in \bar{H}$. Property 2 cannot occur, according to lemma 7 and corollary 1. The lemma is proved.

Lemma 9. If $M(p) > M(q)$, then $H = N[N(p): N(p) + |H| - 1]$.

Proof. From definition 4 and corollary 1, we find $M(p) = \min \{M(u): u \in H\}$. Let us consider two of these peaks--v and w--such that v belongs to the hammock H and $M(p) < M(w) < M(v)$. Using lemmas 1 and 7, we find the existence of the M-path from p to w. It is clear that peak q cannot belong to this path since $M(q) < M(p)$ and accordingly, according to definition 4, $w \in H$. Thus, $H = M[M(p): M(p) + |H| - 1]$ and there exists the M-path from peak p to any peak of H. This completes the proof of the lemma according to definitions 4 and 15. The lemma is proved.

Lemma 10. If $M(q) < M(p)$, then for any $i \in [0: n]$ we have $H = L_i[L_i(p): : L_i(p) + |H| - 1]$.

Proof. Let us carry out the proof by induction with respect to i. Let $i = n$. If for some $j \in [1: n - 1]$, $H = K_j[K_j(p): K_j(p) + |H| - 1]$ is fulfilled, then in view of lemmas 3 and 6 and definition 18, $H = K_{j+1}[K_{j+1}(p): : K_{j+1}(p) + |H| - 1]$ is fulfilled. Hence follows the validity of the lemma for the case $i - n$ since, according to lemma 9, $H = N[N(p): N(p) + |H| - 1]$.

Let us assume that the lemma is valid for all $i \in [s :n]$ and let us consider the case when $i = s - 1$. It follows from lemmas 4 and 6 that either $\bar{H} \subset \subseteq L_s[s]$ or $L_s[s] \subseteq H$ or the intersection $H \cap L_s[s]$ is empty. Thus, it is sufficient to consider the case when $\bar{H} \subset L_s[s]$. In this case either all peaks of H have the same L_s-rank or the L_s-rank of any peak of H differs from the L_s-rank of any peak of $L_s[s] \setminus H$ and is equal to $L_s(w)$ for some $w \in H$. This means that also here $H = L_{s-1}[L_{s-1}(p): L_{s-1}(p) + |H| - 1]$. The lemma is proved.

Lemma 11. If $M(p) < M(q)$, then $\bar{H} = L[p: q]$.

Proof. Let us show by induction with respect to i the validity of the more general statement for the case when $M(p) < M(q)$: for any $i \in [0: n]$, if $i < L_i(p)$, then $\bar{H} = L_i[p: q]$, otherwise $L_i[p: q] \subseteq \bar{H} \subseteq L_i[L_i(p): n]$.

Let $M(p) < M(q)$. Let us consider the case when $i = n$. According to lemma 8, $N[p: q] \subseteq \bar{H} \subseteq N[N(p): n]$. It follows from lemmas 3 and 6 and definition 18 that if $K_j[p: q] \subseteq \bar{H} \subseteq K_j[K_j(p): n]$ for some $j \in [1: n - 1]$, then $K_{j+1}[p: q] \subseteq \bar{H} \subseteq K_{j+1}[K_{j+1}(p): n]$. Thus, the proved statement is valid when $i = n$.

Let us assume that the statement is valid for all $i \geq s$ and let $i = s - 1$. Let us consider two cases: 1) $L_{s-1}(p) > s - 1$ and 2) $L_{s-1}(p) \leq s - 1$.

Let $L_{s-1}(p) > s - 1$. Hence, we find from lemmas 4 and 6 that it is sufficient to consider the case when $\bar{H} \subseteq L_s[s]$. But in this case either all the peaks of $\bar{H}$ have the same L_s-rank or the L_s-rank of each peak $v \in \bar{H}$ coincides with $L_s(w)$ for some $w \in \bar{H}$ and differs from the L_s-rank of any peak of $L_s[s] \setminus \bar{H}$ if $v \neq q$. Thus, in this case $\bar{H} = L_{s-1}[p: q]$. Let $L_s(p) = s$. Then according to the inductive premise $L_s[p: q] \subseteq \bar{H} \subseteq L_s[L_s(p): n]$ and, according to lemmas 4 and 6, $\bar{H} \subseteq L_s[p]$. Hence we find from definitions 4 and 13 that the L_s-rank of any peak $v \in H$ is less than the L_s-rank of peak q and if the L_s-rank of some peak $v \in L_s[p]$ is less than the L_s-rank of peak q, then $v \in H$. Thus, in view of definition 19, we also find $\bar{H} = = L_{s-1}[p: q]$.

Let $L_{s-1}(p) \leq s - 1$. Then according to definition 19 and the inductive premise, $L_s(p) < s$ and $L_s[p: q] \subseteq \bar{H} \subseteq L_s[L_s(p): n]$. From $L_s(p) < s$, according to lemmas 4 and 6 and definition 19, we find that either $L_s(v) = L_{s-1}(v)$ for any peak $v \in \bar{H}$ or the following property is fulfilled for any peak $v \in H$: if $L_s(v) \neq L_{s-1}(v)$, then $v \in H \setminus \{p\}$ and there exists the peak w of H distinct from p such that $L_s(v) = L_{s-1}(w)$. Thus, $L_{s-1}[p: q] \subseteq \bar{H} \subseteq \subseteq L_{s-1}[L_{s-1}(p): n]$. The lemma is proved.

Let S be some subgraph of graph G and let p be some peak of it. Let us determine the following sets: SET1 $(S, p) = \{v \in SUCCESS\ (w): w \in S$ and $M(v) < M(p)\}$; SET2 $(S, p) = \{v \in SUCCESS\ (w): w \in S, W \neq p$ and $L(v) < L(p)\}$; SET3 $(S, p) = \{v \in SUCCESS\ (w): w \in S, M(v) > M(p)$ and $L(v) < L(p)\}$. Let us also introduce MAX1 $(S, p) = \max (L, (p), \{L(v): v \in SUCCESS\ (w), w \in S$ and $M(v) > M(p)\})$; MAX2 $(S, p) = \max (L(p), \{L(v): v \in PRED\ (w), w \in S$ and $w \neq p\})$.

Theorem 7. Some subgraph H of graph G is a hammock with initial peak p and final peak q if and only if the following properties are fulfilled:

1) $H = L[L(p): L(p) + |H| - 1]$;

2) $M(v) > M(p)$ for some peak v of H;

3) the set SET2 (H, p) is empty;

4) the set SET3 (H, p) is empty;

5) $p \bar{\in}$ SUCCESS (q);

6) MAX2 $(H, p) < |H| + L(p)$;

7) either SET1 (H, p) and MAX1 $(H, p) = L(p) + |H| = L(q)$ is empty or SET1 $(H, p) = \{q\}$ and MAX1 $(H, p) < L(p) + |H|$.

Proof. For any hammock H the indicated properties are fulfilled in view of definition 4, lemmas 10 and 11 and corollary 1.

Let some subgraph H satisfy properties 1-7. It follows from properties 3 and 6 that H has a single initial peak p. Property 7 means that H has a final peak q for which either $L(q) = L(p) + |H|$, if SET1 (H, p) is empty, or $q \in$ SET1 (H, p). Let us assume that there exists a final peak v distinct from q for H. We find from property 7 that $M(v) > M(p)$, $L(v) < L(p)$ and consequently $v \in$ SET3 (H, p). But SET3 (H, p) is empty according to property 4. Consequently, a contradiction occurs. Thus, H is a hammock with initial peak p and final peak q.

The theorem is proved.

Corollary 7. Let p be some peak of the graph and S be the subgraph $L[L(p): L(p) + k]$, $k \geq 0$, for which at least one of the following properties is fulfilled:

1) SET2 (S, p) is not empty;

2) SET3 (S, p) is not empty;

3) there is the peak v in S such that $M(v) < M(p)$;

4) $L(p) + k = n - 1$.

The hammock H with intial peak p does not then exist such that $S \subset H$.

Let us now describe the fourth step of the hammock separation algorithm. Completion of this step consists of n - 1 steps, in which all hammocks having initial fixed peak p as the initial peak are found in each of the steps. This separation is accomplished in the following manner. All the predecessors of peak p are initially noted. The sets of the enumerated peaks L[L(p): L(p) + i], beginning with the set for which i = 0, are then enumerated sequentially. To determine whether the current set of hammocks is formed, they are calculated for p and the given set SET1, SET2, SET3, MAX1 and MAX2. The conditions of theorem 7 and corollary 7 are checked after the indicated calculations. Transition to consideration of the next set is then made as a function of the results of this check or finding the hammocks with given initial peak is completed. Realization of a single step of the indicated part of the algorithm is presented below:

```
procedure HAMMOCKS (L, M, p)
    MAX1: = MAX2: = L(p); SET1: = truth
    take peak p as q
    H is an empty set
    imbed each peak v of PRED (p)
    for k from L(p) to n - 1 through 1 cycle
        add peak v to H for which L(v) is equal to k
        for all w from SUCCESS (v) cycle
            if M(w) < M(p)
            then if SET1 or w = - this q
                then SET1: = false
                    take peak w as q
                otherwise complete finding hammocks,
                    for which p is the initial peak
                all
            otherwise if L(w) > L(p)
                then MAX1 is the maximum of MAX1 and
                L(w)
                otherwise complete finding the hammocks for which p
                is the initial peak
                all
            all
        all
            if v is not p
            then for all w of PRED (v) cycle
                    if L(w) < L(p)
                    then complete finding the hammocks,
                         for which p is the initial peak
                    otherwise MAX2 is the maximum of MAX2
                    and L(w)
                    all
```

```
                all
            all
            if ¬ SET1
            then if MAX1 < k + 1 and q is not noted
                then H is a hammock with initial peak p
                    and final peak q
                all
                complete finding the hammocks for which
                p is the initial peak
            otherwise select the peak u for which L(u)
                is equal to k + 1
                if MAX1 = k + 1 and u is not noted
                then H is a hammock with initial peak p
                    and final peak u
                all
            all
        all
end
```

Each step of this part of the algorithm occupies time O(m) since any peak of the graph can be processed no more than once, while each processing is related to single inspection of the predecessors and successors of the peak. Thus, the laboriousness of the given part of the algorithm also comprises O(nm) and consequently O(nm) is the total estimate of the laboriousness of the hammock separation algorithm.

Conclusions

The given algorithms not only have good time complexity by order, but are also not very complicated to realize, have small proportionality constants and therefore can be used in practice. The graph can be represented by three files in Algol-realization of the indicated algorithms: PEAKS [1: n, 1: 2], PRED [1: n + m] and SUCCESS [1: n + m]. The variables PEAKS [i, 1] and PEAKS [i, 2] contain the names of the first predecessor and the first successor of the peak with name i (the names of the objects are understood as their numbers in the corresponding files). The set of names of the peak-predecessors (and of the peak-successors) of any peak is arranged in the appropriate file sequentially and is completed by a zero. Thus, storage of the graph requires 2m + 4n memory cells. Each index F is two files: F-NUMBERS [1, n] and PEAKS WITH RESPECT TO F-NUMBERS [1: n]. Information about path P which is constructed during N-index is stored in two files--FLOW1 [1: n] and FLOW2 [1: n] in which the names of the peaks of the path and the first of their inspected successors are stored at each moment of time. With this representation, the total memory required for the working data of each of the two algorithms does not exceed 8n + 20 cells. If one disregards consideration of the memory required to store the separated subgraphs, then each algorithm requires no more than 2m + 12 N + + 20 memory cells.

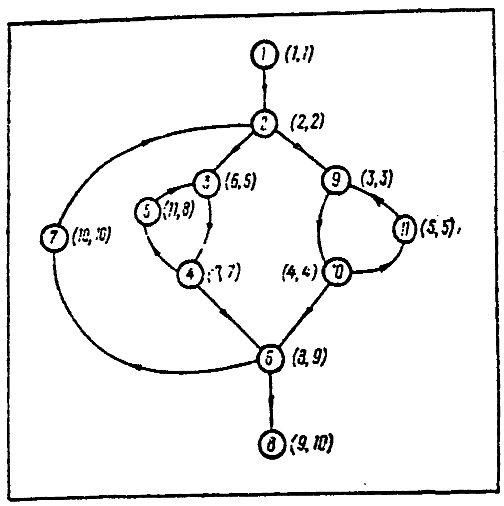


Figure 5. Graph and Its Possible N and L Indices

BIBLIOGRAPHY

1. Allen, F. E., "Method of Determining Information Relationships in Programs," in "Teoriya programmirovaniya" [Programming Theory], Part 2, Novosibirsk, 1972.

2. Acho, A. and J. Ullman, "Teoriya sintaksicheskogo analiza, perevoda i kompilyatsii" [The Theory of Syntactical Analysis, Translation and Compilation], Part 2, Moscow, Mir, 1978.

3. Acho, A., J. Hopcroft and J. Ullman, "Postroyeniye i analiz vychislitel'nykh algoritmov" [Construction and Analysis of Calculating Algorithms], Moscow, Mir, 1979.

4. Val'kovskiy, V. A. and V. N. Kas'yanov, "Large-Block Segmentation and Parallelling of Program Schemes," PROGRAMMIROVANIYE, No 1, 1976.

5. Yershov, A. P., "Universal Programming Processor," in "Problemy prikladnoy matematiki i mekhaniki" [Problems of Applied Mathematics and Mechanics], Moscow, Nauka, 1971.

6. Koch, U., "The Global Economy of Instructions," in Proceedings of the Second All-Union Conference on Programming. Reports of Foreign Participants, Novosibirsk, 1970.

7. Kas'yanov, V. N., "Analysis of Program Control Graphs," in "Sistemnoye programmirovaniye" [Systems Programming], Part 2, Novosibirsk, 1973.

8. Kas'yanov, V. N., "Separation of Hammocks in an Oriented Graph," DOKLADY AN SSSR, Vol 221, No 5, 1975.

9. Kas'yanov, V. N. and M. B. Trakhtenbrot, "Analysis of Program Structures in Global Optimization," in "Metody realizatsii novykh algoritmicheskikh yazykov" [Methods of Realizing New Algorithmic Languages], Part 1, Novosibirsk, 1975.

10. Kas'yanov, V. N., "Relief of Sections of Repetition," PREPRINT OF VTS SO AN SSSR, No 178, Novosibirsk, 1979.

11. Pottosin, I. V., "The Problem of 'Purification of Cycles,'" in TSIFROVAYA VYCHISLITEL'NAYA TEKHNIKA I PROGRAMMIROVANIYE, No 4, Moscow, Sovetskoye Radio, 1968.

12. Pottosin, I. V., "Program Linearization and Partial Ordering of Programs," in "Sistemnoye i teoreticheskoye programmirovaniye" [Systems and Theoretical Programming], Novosibirsk, 1972.

13. Pottosin, I. V., "Global Optimization: A Practical Approach," in "Metody realizatsii novykh algoritmicheskikh yazykov," Part 1, Novosibirsk, 1975.

14. Sabel'feld, V. K., "Realization of Procedures in a Multilanguage Translator," in "Metody realizatsii novykh algoritmicheskikh yazykov," Vol 1, Novosibirsk, 1975.

15. Allen, F. E., "Program Optimization," in ANNUAL REVIEW IN AUTOMATIC PROGRAMMING, Vol 5, New York, Pergamon Press, 1969.

16. Earnest, C. P., "Some Topcis in Code Optimization," JOURNAL OF THE ACM, Vol 21, No 1, 1974.

17. Earnest, C. P., K. G. Balke and J. Anderson, "Analysis of Graphs by Ordering of Nodes," JOURNAL OF THE ACM, Vol 19, No 1, 1972.

18. Hecht, M. S., "Topological Sorting and Flow Graphs," in Proceedings of IFIP, Congress 74, TA-3, Amsterdam: North Holland Publishing Company, 1974.

19. Hecht, M. S. and J. D. Ullman, "A Simple Algorithm for Global Data Flow Programs," SIAM J, COMPUTING, Vol 4, No 4, 1975.

20. Hecht, M. S., "Flow Analysis of Computer Programs, New York, 1977.

21. Kam, J. B. and J. D. Ullman, "Monotone Data Flow Analysis Frameworks," ACTA INFORMATICA, Vol 7, No 3, 1977.

22. Kas'yanov, V. N., "Some Properties of Fully Reducible Graphs," INFORMATION PROCESSING LETTERS, Vol 2, No 4, 1973.

23. Kas'yanov, V. N. and S. B. Pokrovsky, "Schematology in a Multi-Language Optimizer," in LECTURE NOTES IN COMPUTER SCIENCE, Vol 53, Berlin-Heidelberg-New York, Springer-Verlag, 1977.

24. Tarjan, R. E., "Depth First Search and Linear Graph Algorithms," SIAM J, Vol 1, No 2, 1972.

25. Ullman, J. D., "Fast Algorithm for the Elimination of Common Subexpression," ACTA INFORMATICA, Vol 2, No 3, 1973.

[8044/0004B-6521]

6521
CSO: 8044/0004B

UDC 681.324

ON ONE CLASS OF PROBLEMS ON THE OPTIMUM DISTRIBUTION OF COMPUTER FACILITY RESOURCES

Kiev PROGRAM.-TEKHN. KOMPLEKSY REAL'N. VREMENI [Real-Time Programmed Technological Complexes] in Russian 1979 pp 116-121

GAVRILOVA, N.L., MOROZOV, A.A., and TSYTSURIN, V.Ye.

[From REFERATIVNYY ZHURNAL, AVTOMATIKA, TELEMEKHANIKA I VYCHISLITEL'NAYA TEKHNIKA No 7, 1980 Abstract No 7B97 by T.M. Kuznetsova]

[Text] The authors discuss a class of systems, the operating mode of which is determined by a criterion and restrictions that do not depend on the order of solution of individual problems. Examples of such systems are the simultaneous solution of a group of problems that are independent of each other. The authors solve the problem of the distribution of the resources of a multimachine complex of computer facilities and an optimum (according to the criterion of cost) complex of computer facilities. The problems is interpreted as the minimax problem of the optimum (according to the criterion of operating speed) computational process and is reduced to an integral problem in linear programming. When all the machines in a multicomputer complex are operating simultaneously, the time of completion of the computation process is determined by the operating time of the computer that is in operation longer than all the others. In connection with this, the optimum solution consists of that distribution of the work that minimizes this time. The optimization method proposed by the authors consists of searching for an initial approximation with the help of a preliminary solution of the intermediate linear programming problem, with subsequent variation of the order of performance of the work in order to find the solution with the minimum machine time losses. As an example the authors discuss the problem of the optimum (according to operating speed) distribution of the resources of a complex consisting of three processors, in which six types of information processing are performed. The use of their proposed approach makes it possible to reduce the machine time required to search for the optimum (according to operating speed) resource distribution by a factor of five. References 4.

[22-11746]

UDC 681.324

EVALUATING THE EFFICIENCY OF THE USE OF MULTIPROCESSOR COMPUTERS FOR THE REALIZATION OF COMPLEX ALGORITHMS FOR CONTROL SYSTEMS

Kiev PROBL. GIBRID. VYCHISL. TEKHN. [Problems of Hybrid Computer Technology] in Russian 1979 pp 30-38

GANITULIN, A.Kh., and POLYAKOV, G.A.

[From REFERATIVNYY ZHURNAL, AVTOMATIKA, TELEMEKHANIKA I VYCHISLITE'NAYA TEKHNIKA No 7, 1980 Abstract No 7B114]

[Text] The authors explain a method for evaluating the efficiency of the use of multiprocessor computers of different types for the realization of complex, parallel algorithms for control systems. In their explication they use a two-stage approach to evaluating efficiency: in the first stage the efficiency indices are determined for each individual algorithm, while in the second stage they evaluate the efficiency of the utilization of multiprocessor computers for the realization of a complex algorithm on the basis of the results obtained during the execution of the first stage. In both stages they recommend simulation models of the algorithm realization processes that make it possible to obtain quantitative values of the efficiency indices (average realization time, computer reliability and work load with due consideration for the characteristics of the computer's structure, special features of the algorithms, and the conditions under which the computer actually functions). Figures 1; references 5.

[22-11746]

UDC 681.322.06.015

OPTIMIZATION OF THE 'RELIABILITY' PARAMETER IN REAL-TIME SYSTEMS

Kiev PROGRAM.-TEKHN. KOMPLEKSY REAL'N. VREMENI [Real-Time Programmed Technological Complexes] in Russian 1979 pp 16-20

ANDROSENKO, S.G., and TIMOFEYEV, A.B.

[From REFERATIVNYY ZHURNAL, AVTOMATIKA, TELEMEKHANIKA I VYCHISLITEL'NAYA TEKHNIKA No 7, 1980 Abstract No 7B168 by T.M. Kuznetsova]

[Text] The authors discuss questions related to the development of devices, the structure of which is based on new methods that, because of the programmed rearrangement of the structure, give flexibility to the parameters of reliability, effective speed and the number of terminals that can be serviced. This makes it possible to maintain the reliability parameter in the system at a given level at all times and to give the information processing unit a full work load. Flexibility is achieved with the help of programmed (internal) and modular (external) adjustments of the structure; the latter action consists of replacing some modules with others. In order to control the reliability parameter's level, the device's structure contains a method that is a modification of retrospective analysis and is based on the determination of some permissible deviation of a received signal from the ideal. The effectiveness of the method has been checked by a comparative analysis of the probabilities of the nondetection of errors and the noncorrection of errors in a byte and in a unit for classical methods and the new method. The results of the comparison, which are presented in a table, indicate that when the new method is used, there is an improvement in correcting ability by a factor of 10-100 for the communication channels that are most common in computer networks, where the information processing points are quite distant from each other. When the intensity of the interference is reduced, the proposed method makes it possible to obtain an even greater gain in reliability. Figures 1; references 1.

[22-11746]

UDC 681.322.019

SYSTEM FOR CONTROLLING A RELATIONAL DATA BASE ON A MINICOMPUTER

Vladivostok IN-T AVTOMAT. I PROTSESSOV UPR. S VYCHISL. TSENTROM DAL'NEVOST. NAUCH. TSENTR AN SSSR. PREPRINT [Institute of Automation and Control Processes, Computer Center, Far Eastern Scientific Center, USSR Academy of Sciences: Preprint] in Russian No 3, 1980, 29 pages

OLENIN, M.V., and SEMENOV, S.M.

[From REFERATIVNYY ZHURNAL, AVTOMATIKA, TELEMEKHANIKA I VYCHISLITEL'NAYA TEKHNIKA No 7, 1980 Abstract No 7B186]

[Text] The authors describe a data base control system that has been developed for minicomputers of the M400 type (SM3, SM4). The inquiry language is based on the calculus of relations and is close in capabilities to (Kodd's) language DSL-α. The authors present an algorithm for the realization of the inquiry language, as well as the basic control tables for the system and the structure of its software. They also describe facilities for creating and editing relationships and the organization of the search in a data base. Figures 4; references 3.

[22-11746]

APPLICATIONS

UDC 681.3

ELEKTRONIKA-60 SIMULATION SYSTEM

Kiev ELEKTRONNOYE MODELIROVANIYE in Russian No 5, 1980 signed to press 27 Aug 80 p 103

[Article by V. Z. Lyakhovich, E. L. Onishchenko, V. N. Goncharov and S. M. Skorobogat'ko]

[Text] A hardware complex for electronic simulation of production processes and control of dynamic objects with analog actuating members, specifically, to simulate power engineering objects, rolling and chemical plant processes and for simulation in the mining industry, has been developed at the Kiev Automatic Equipment Institute. A subsystem for control of reagent proportioning in the potassium chloride flotation process, an external view of which is presented on the second page of the cover, was constructed on the basis of this complex. The subsystem processes information coming in from 12 sensors and controls six acutating mechanisms [1].

The complex contains a computer constructed on the basis of the Elektronika-60 microcomputer and peripheral equipment (a Consul-260, FS-1501, PL-150 and devices for communicating with the object).

The specifications of the calculating part of the complex are determined by the capabilities of the Elektronika-60 microcomputer [2] and the serial modules which supplement it: P1--4K internal storage of 16-digit words, PP1 and 15UZPP--reprogrammable permanent storage with capacity of 4K and 2K words, respectively, and I1--a communications interface. The machine has developed punch tape software which includes an assembler, BASIC and also standard and supplementary programs.

The device for communicating with the object contains the following modules: a timer, signal converter and interface.

The timer module emits signals which control USO [device for communicating with the object] operation and organizes the operating cycles lasting up to 24 hours with sampling up to 1 ms if needed.

The signal-conversion modules consists of 0-5 mA analog signal converter, 0-1 kHz frequency signal converter and binary-decimal to binary eight-digit code converter. Moreover, the eight-digit binary code is converted to a 0-5 mA analog signal.

The interface module for computer communication with the USO provides access of the computer through the channel to the sensor registers and the actuating mechanisms. Addresses from the memory field of the machine are assigned to the registers in this case.

The USO is designed on the basis of the ASVT-M on microcircuits of series K133, K140, K155, K252 and K589.

BIBLIOGRAPHY

1. Sukhomlinov, M. M., V. Z. Lyakhovich and P. A. Voloshin, "Ispol'zovaniye mikro-EVM v ASU TP" [Using Microcomputers in Automated Control Systems for Production Processes], Kiev, Znaniye, 1979.

2. Borisenko, V. D., V. V. Plotnikov and I. L. Talov, "The Elektronika-60 Microcomputer," ELEKTRONNAYA PROMYSHLENNOST', Moscow, No 10, 1978.

[427-6521]

6521
CSO: 1863

ALL-UNION SEMINAR 'HOMOGENEOUS CALCULATING STRUCTURES AND SMALL COMPUTERS'

Kiev ELEKTRONNOYE MODELIROVANIYE in Russian No 5, 1980 signed to press 27 Aug 80 p 105

[Article by Ye. V. Babichev and T. I. Zrelov]

[Excerpts] The All-Union Seminar "Homogeneous Calculating Structures and Small Computers" was held in Moscow in December 1979. A total of 180 specialists (including 8 doctors and 44 candidates of technical sciences) in the field of computer technology, representing scientific research organizations and production associations from 24 cities of the country, participated in the work of the seminar.

Problems related to the structure and configuration of small computers, software and use of them and also problems related to development and use of homogeneous structures were discussed at the seminar. A total of 56 reports, 10 of which were presented in the form of plenary talks and the remaining ones were presented in the form of reports from the podium, was heard. The plenary reports presented a survey of the existing state of the art and the prospects for development of individual directions of the topics under discussion.

The future trends for development and use of small and microcomputers in our country and abroad were given in the report of I. V. Prangishvili.

New principles of organizing homogeneous structures were outlined and problems of optimum disposition of resolving and switching components were outlined in A. V. Kalyayev's report.

P. P. Parkhomenko gave a survey of the current state of the art of the theory of technical diagnosis and pointed out timely unresolved problems which arise with development of microelectronic equipment.

Problems of organizing multimachine complexes, problems of communications between processors in the same machine and communications between machines in computer networks were considered in V. G. Lazarev's report.

S. Ya. Vilenkin's report was devoted to organization of the calculating process in a computer system with a single flow of instructions and a multiple flow of data.

V. V. Ignatushenko talked about organization of multiprocessor computer systems and about a new approach to solution of problems of large dimensionality using teaching methods.

[427-6521]

6521
CSO: 1863

EFFECTIVENESS OF HIGH RELIABILITY OF EQUIPMENT

Moscow VOPROSY EKONOMIKI in Russian No 7, 1980 pp 60-68

[Article by A. Konson, Leningrad]

[Excerpt] The most effective alternative of equipment reliability is that which assures a minimum of reduced annual expenditures dependent on reliability. If the economic effect from improvement of equipment reliability consists not in a saving of expenditures but in better satisfaction of some sort of qualitative requirement of society (for example, improvement of the quality of television broadcasting because of the elimination of interruptions caused by an unforeseen failure of the system), then for economic evaluation the indicator of relative effectiveness of additional expenditures connected with increase of reliability (1) is:

$$\overset{(1)}{\varepsilon_0} = \frac{\Delta P_{\text{д}}\,(2)}{\Delta W_{\text{г.д}}\,(3)}, \qquad (1)$$

where (2) is the additional increase of reliability of the system as a percentage of its initial value; (3) is the relative increase in the reduced annual expenditures to increase the reliability of the system as a percentage of their initial value.

The most effective means of increasing equipment reliability is increasing that of its separate parts and components, which can involve increasing their cost. However, in all types of equipment designed for long-term operation an attempt is made to use elements with high reliability, even if they are more costly. In that case special attention is given to potentially critical parts and assemblies. When units with high reliability are used in critical assemblies of equipment, its dependability can be considerably increased without providing spare parts. To increase the reliability of parts or units, developers strive to ease their working conditions by reducing the voltage, current, temperature or mechanical loads. In many cases this permits reducing the failure rate by nine tenths.

Automation of the assembly of hybrid integrated circuits permits not only increasing the productivity of assembly equipment but also increasing assembly precision and the reliability of a hybrid film unit. High-speed multi-purpose integrated circuits have been worked on since 1979. They will contribute to a considerable increase of dependability of very critical electronic systems, and also to a saving of expenditures on their servicing and maintenance with a simultaneous reduction of dimensions, mass and power consumption and the possibility of the creation of "intellectual systems." A considerable increase of reliability of systems based on high-speed multi-purpose integrated circuits will be achieved as a result of reduction of the number of units and interconnections.

The saving of the reduced annual expenditures obtained from increasing the reliability of a separate part replaced when it has failed (1), in rubles per year per part, is determined as:

$$\underset{(1)}{\Delta W_{г.э}} = \underset{(2)}{\Delta S_{з.э}} + \underset{(7)}{\varepsilon_н} \underset{(3)}{\Delta F_э} + \underset{(4)}{\Delta U_{п.э}} - \varepsilon_н (\underset{(5)}{z_{л2}} - \underset{(6)}{z_{л1}}), \qquad (2)$$

where (2) is the annual saving of resources connected with replacement of that part; (3) is the saving of resources connected with the user's stock of the part; (4) is the saving resulting from reduction of damage due to downtime of equipment in repairs because of failure of the given part; z is the price of a part with low (5) and high reliability (6); (7) is the normative coefficient of effectiveness of additional capital investments.

In that case the annual saving, not only on the cost of the part itself per year of use but also of expenditures connected with its replacement, is taken into consideration in the value of (1) in formula (3) if the given part has been placed more conveniently in the design, in rubles per year per part:

$$\Delta S_{з.э} = \frac{\overset{(2)}{z_{л1}} + \overset{(4)}{B_{з1}}}{T_{с1}} - \frac{\overset{(3)}{z_{л2}} + \overset{(5)}{B_{з2}}}{T_{с2}}, \qquad (3)$$

where $B_з$ are the basic and supplementary wages of a worker replacing the given part, together with charges on the wages for social insurance; (2) and (3) are the prices of parts with low and high reliability respectively; (4) and (5) are the wages in a previous, less successful arrangement of the part in the design and in its new, more successful arrangemnt respectively; (6) and (7) are the service lives of parts with low and high reliability respectively, in years.

Taking damages resulting from downtime of equipment during repairs into account in the economic calculation leads to a need to connect its separate element with the entire system it is built into. In systems by means of which production is produced, freight is transported or information is transmitted, sudden failure can cause underproduction, a reduction of freight shipments or undertransmission of information. If the change of system reliability affects the length of downtime in preventive and unplanned current repairs by changing the annual productivity of the system, it is necessary to determine the damage caused by downtime in those and other repairs. For large costly systems (ships, aircraft, electric power systems and metallurgical plant equipment) damage resulting from downtime in all types of repair, including capital repairs, must be taken into consideration. At the same time, for cheaper equipment, the normal loading of which is not great, one can limit oneself to consideration of the damage resulting from downtime only in unplanned current repairs.

One way to increase equipment reliability is to install duplicating elements at the most critical points. If elements are connected by means of switches, then if one of them fails the spare can be switched on. In that case the detection of defects and the switching of circuits from failed elements to spares in good working order must be highly reliable and be completed without dalay. Ultrareliable equipment can be created by providing multiple spares, but then there will be high operating expenditures, which lead to a general increase of expenses. Providing spares is completely necessary in cases where it provides prevention of accidents due to equipment failure. In that case multiple spares are used if sudden equipment failures can lead to deaths of human beings. In many other cases, however, the effect from providing equipment with spares consists in reducing damage due to downtime in repairs. At the same time, providing spares can cause a considerable increase in the cost of that equipment, greater expenditures on repairs, additional power consumption and also the storage of a larger number of parts and units by the user.

The saving of reduced annual expenditures obtained from providing spare parts for equipment (1), in rubles per year per system, amounts to:

$$\underset{(1)}{\Delta W_{\text{г.э}}} = \underset{(2)}{\Delta U_{\text{п.э}}} - [\underset{(4)}{(a_{\text{н}} + \varepsilon_{\text{н}})} \underset{(3)}{\Delta z_{\text{д}}} + \underset{(7)}{\Delta R_{\text{т.д}}} + \underset{(5)}{\Delta S_{\text{эл.д}}} + \underset{(6)}{\varepsilon_{\text{н}} \Delta F_{\text{д}}}], \quad (4)$$

where (2) is the reduction of damage resulting from downtime of equipment during repairs; (3) is the additional price of equipment with spare parts; (4) is the annual standard of depreciation deductions for that equipment; (5) is the additional electric power consumed by back-up parts; (6) is the additional store of spare parts of equipment at the user's for its back-up equipment; (7) is the expenditures on current repairs of back-up equipment.

Providing spare parts must be examined in combination with the use of high-reliability parts in equipment. Those variants do not exclude, but complement one another. Therefore even before considering the variants of providing spares it is important to determine whether all the economically effective means of increasing the reliability of separate parts have been used in that equipment. Different variants of providing spare parts are usually examined. The selection of the most effective of them will depend on the required level of reliability and the reduced annual expenditures.

In recent years much attention has been given to the reduction of expenditures on current repairs and reduction of damage due to downtime of equipment in current repairs. To do that they have begun to build into it automatic devices which permit reducing expenditures on repairs through a saving fo time spent on finding defects, and also as a result of full use of the possibilities of each part and elimination of their premature replacement. The increasing complexity and cost of many types of present-day equipment make it advantageous to establish strictly fixed service lives of parts and assemblies, with obligatory replacement regardless of their actual condition.

To reduce expenditures on the current repairs, parts and assemblies are often replaced when their actual condition requires it. To obtain selective information about the condition of separate very critical parts and assemblies during their operation, devices have begun to be created which permit monitoring wear and simultaneously discovering forming defects. They permit reducing expenditures on repairs not only through reduction of the time needed for the detection of defects but also thanks to the fact that the parts and assemblies are not removed from the equipment ahead of time, as was the case when their replacement periods were fixed. The fact that the automatic monitoring of condition rids the system of excessive dismantling also is reflected in a reduction of expenditures on repairs and of damage from downtime in repairs.

Devices for automatic defect monitoring and detection often are fairly complex and so the price of the system increases. In addition, growth of the number of elements making up the system increases the frequency of its failures. When automatic defect monitoring and detection devices are used one must be convinced that the obtained advantages justify the additional expenditures.

The saving of reduced annual expenditures obtained from the use of automatic defect monitoring and detection devices (1), in rubles per year per system, amounts to:

$$\underset{(1)}{\Delta W_{п.э}} = (\underset{(2)}{\Delta R_{т.э}} + \underset{(3)}{\Delta U_{п.э}}) - [\underset{(4)}{(a_н + \varepsilon_н)} \underset{(5)}{z_к} + \underset{(6)}{\Delta R_{т.д}} + \underset{(7)}{\varepsilon_н \Delta F_д} + \underset{(8)}{\Delta S_{эл.д}}], \quad (5)$$

where (2) is the annual saving of expenditures on current repair of the system resulting from very rapid discovery of defects and an absence of premature replacement of parts; (3) is the saving because of reduction of damage from downtime of the systems in repairs thanks to very rapid defect detection; (4) is the annual standard of depreciation deductions on defect monitoring and detection equipment; (5) is the price of special defect monitoring and detection equipment built into the system; (6) is the additional annual expenditures on the repair of defect monitoring and detection apparatus; (7) is the cost of electric power additionally consumed annually by the defect monitoring and detection apparatus; (8) is the additional store of replaceable parts for special defect monitoring and detection apparatus stored at the user's, in rubles.

Also of great importance is the correct selection of protective devices, for example, protective apparatus which shuts off the system during damage or when a dangerous operating condition develops. Although disconnecting protective apparatus is not always capable of completely preventing system damage, it reduces its scope and by the same token the cost of repairs and the damage from downtime of the system in repairs. At the same time the more complex the protective devices the greater the probability of their failure and the more substantially they increase the cost of the system. On the basis of technical and economic analysis it is necessary therefore to determine how in each specific case to correctly select the types of protective devices.

The gain from the use of protective devices consists in the saving because of reduction of the damage resulting from accidents of the system in the course of the year in the absence of protection. Form that saving, obtained as a result of the use of protective devices, in the calculation of the saving of reduced annual expenditures it is necessary to deduct: the product of the price the productive times the annual standard of depreciation deductions and times the normative coefficient of effectiveness of capital investments; expenditures on additional current repairs of the protective device under consideration in the course of the year; the cost of electric power additionally consumed annually by the protective device; expenditures connected with additional damage resulting from downtime of the system due to repairs of the protective device in the course of the year.

Also of great importance in providing high reliability is the correct establishment of the nomenclature, quantity and places of storage of spare parts for equipment.

Thus, most often when an integrated circuit fails it is replaced. The defective plate is sent to the plant for repair. This occurs because there are not sufficiently qualified personnel for repairs at the place of operation. At the same time the replacement of integrated circuits which have become defective gives rise to a need to always have a sufficient quantity of spare parts at the place of operation. At the present time, according to some estimates, a stock of circuit plates worth 6 billion dollars has accumulated in the world in warehouses and repair workshops [1].

Determination of the optimum stock of replaceable parts stored at the user's (1), in rubles per year per system, is based on observance of the condition:

$$\varepsilon_{н}\Delta F_{д}<\Delta U_{п.з.} \qquad (6)$$

(1) (2) (3)

where (2) is the cost of the additional stock of replaceable parts for a system, stored at the user's, in rubles; (3) is the saving from reduction of damage because of shortening of downtimes due to lack of spare parts.

With reference to digital systems a new problem has arisen in recent years--the creation of failure-resistant systems capable of recommencing work according to a definite program upon the appearance of a defect. Failure resistance permits a logical machine to continue work also in cases when various component failures arise in the system within which it works. By 1978 several prototype failure-resistant systems had already been created. The failure resistance of digital computers is assured by means of internal provision of spare parts with a "supplementary logic." the presence of a module for error processing and codes for error detection and correction, the creation of redundancy to monitor internal circuits and of self-monitoring modules to check errors.

Failure-resistant electronic digital computers are intended above all for systems whose failure can threaten human life (for example, an air traffic control system and the control of trains and aircraft), inflict great financial harm (for example, failure of a large computer system working in an operative regime with time sharing interrupts the work of hundreds of users connected to the system through their terminals) and gives rise to a need for manual technical servicing where possible (for example, satellites on a circumterrestrial orbit and space vehicles in interplanetary flight).

The price of a failure-resistant electronic digital computer increases somewhat because of expenditures on the introduction of properties of failure resistance in the stage of system development and also because of redundant software automatically detecting and eliminating failures. It must be borne in mind, however, that in ultralarge-scale integrated circuits the rate of cost increase diminishes in proportion to the increase in the number of components. As a result of that the introduction of additional circuit elements based on ultralarge-scale integrated circuits to increase the failure resistance of the system increases the cost of electronic digitial computers insignificantly.

The saving of operating expenditures assured by failure resistance of electronic digital computers forms through: reduction of the expenditure of replaceable parts, since the failure of separate components does not worsen the reliability of of the system as a whole; almost complete elimination of downtime during emergency repairs and the damage inflicted on the user of the system by them; reduction of expenditures on planned preventive replacement of unsuitable parts in comparison with the maintenance of a team of specialists in repair of the system who have been permanently on call.

The further development of the problem of economic analysis of equipment reliability will contribute to the creation of highly reliable equipment for the national economy.

BIBLIOGRAPHY

1. ELECTRONICS, No 12, 1978, pp 83-84.

[41-2174]

2174
CSO: 1863

END